Learning Structured Representations for Understanding Visual and Multimedia Data

Alireza Zareian

Submitted in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy
under the Executive Committee
of the Graduate School of Arts and Sciences

COLUMBIA UNIVERSITY

2021

Abstract

Learning Structured Representations for Understanding Visual and Multimedia Data

Alireza Zareian

Recent advances in Deep Learning (DL) have achieved impressive performance in a variety of Computer Vision (CV) tasks, leading to an exciting wave of academic and industrial efforts to develop Artificial Intelligence (AI) facilities for every aspect of human life. Nevertheless, there are inherent limitations in the understanding ability of DL models, which limit the potential of AI in real-world applications, especially in the face of complex, multimedia input. Despite tremendous progress in solving basic CV tasks, such as object detection and action recognition, state-of-the-art CV models can merely extract a partial summary of visual content, which lacks a comprehensive understanding of what happens in the scene. This is partly due to the oversimplified definition of CV tasks, which often ignore the compositional nature of semantics and scene structure. It is even less studied how to understand the content of multiple modalities, which requires processing visual and textual information in a holistic and coordinated manner, and extracting interconnected structures despite the semantic gap between the two modalities.

In this thesis, we argue that a key to improve the understanding capacity of DL models in visual and multimedia domains is to use structured, graph-based representations, to extract and convey semantic information more comprehensively. To this end, we explore a variety of ideas to define more realistic DL tasks in both visual and multimedia domains, and propose novel methods to solve those tasks by addressing several fundamental challenges, such as weak supervision, discovery and incorporation of commonsense knowledge, and scaling up vocabulary. More specifically, inspired by the rich literature of semantic graphs in Natural Language Processing (NLP), we explore innovative scene understanding tasks and methods that describe images using semantic graphs, which reflect the scene structure and interactions between objects. In the first

part of this thesis, we present progress towards such graph-based scene understanding solutions, which are more accurate, need less supervision, and have more human-like common sense compared to the state of the art.

In the second part of this thesis, we extend our results on graph-based scene understanding to the multimedia domain, by incorporating the recent advances in NLP and CV, and developing a new task and method from the ground up, specialized for joint information extraction in the multimedia domain. We address the inherent semantic gap between visual content and text by creating high-level graph-based representations of images, and developing a multitask learning framework to establish a common, structured semantic space for representing both modalities. In the third part of this thesis, we explore another extension of our scene understanding methodology, to open-vocabulary settings, in order to make scene understanding methods more scalable and versatile. We develop visually grounded language models that use naturally supervised data to learn the meaning of all words, and transfer that knowledge to CV tasks such as object detection with little supervision. Collectively, the proposed solutions and empirical results set a new state of the art for the semantic comprehension of visual and multimedia content in a structured way, in terms of accuracy, efficiency, scalability, and robustness.

Table of Contents

List of Figures

List of Tables

Acknowledgements

The achievements reported in this thesis are direct results of Professor Shih-Fu Chang's continuous advice. I admire his outstanding leadership of the Digital Video and Multimedia lab (DVMM), his years of wide-spread success, and his tireless work towards a better future for science and education. Indeed, this thesis is nothing but a collection of collaborative works, involving several renowned researchers and dear colleagues. I especially thank Professor Heng Ji for her invaluable advice through several projects including M^2E^2(Chapter 6), as well as Dr. Manling Li for her diligent work on that project. I also thank Dr. Svebor Karaman for his collaboration in Chapters 3 and 4, along with countless other colleagues who helped me in some way during this journey, including Haoxuan You, Zhecan Wang, Zheng Shou, Spencer Whitehead, Marjorie Freedman, Hanwang Zhang, Brian Chen, Bo Wu, Carl Vondrick, Kai-Wei Chang, Harold Li, Kevin Dela Rosa, Derek Hao Hu, Hang Gao, to name a few.

Besides colleagues and collaborators, my research has always relied on the endless support of the Columbia University staff, especially Computing Research Facilities. Our work was also facilitated by the financial support of various organizations, including DARPA, National Institute of Justice, Mitsubishi Electric Research Labs, and Snap Inc, as well as the NYC Media Lab.

Finally, I thank my wife for her kindness, patience, and support, my mother, for her selfless support and encouragement at every stage of my life, and my father, who has always been my greatest teacher and role model.

To my beloved wife and my dear parents.

Chapter 1: Introduction

> *"spend the summer linking a camera to a computer and getting the computer to describe what it saw"*
>
> — Marvin Minsky on the goal of a 1966 undergraduate summer research project [1]

Deep Learning [2] has achieved remarkable success in the past few years, with exciting potential applications in a variety of sectors. However, despite proficiency in basic tasks such as visual object detection [3], and text translation [4], more complex and realistic functions are still out of reach, such as autonomous driving and Visual Question Answering (VQA). Although large-scale training of deep neural networks has repeatedly broken records on performance leaderboards, quantitative metrics can be misleading and deeper inspection suggests that such models often make serious mistakes that question their understanding ability [5, 6]. This is mostly because mainstream end-to-end deep learning tends to exploit biases in data as shortcuts to greedily minimize the average cost [7, 8, 9]. It is also difficult to interpret how such models make a certain decision, and to verify that the right decision is indeed made for the right reasons [10].

The unreliability, bias, and uninterpretability of neural networks is due to their dependence on black-box, flat, distributed representations of data, which are typically high-dimensional vectors of real-valued numbers. More specifically, these representations are *distributed* as opposed to *symbolic*, which makes them efficient but hard to make sense of, and are *flat* vectors, as opposed to *compositional* structures, which makes them unsuitable for the kind of high-level reasoning that humans do every day. A plausible remedy for such limitations is to explicitly teach neural networks to extract an intermediate symbolic representation from data, which conveys the inherent structure of its semantic content, and exploit that representation to solve any downstream task

that requires reasoning. Natural Language Processing (NLP) has long used semantic graphs to represent raw text [11]. This type of structured, symbolic representation simplifies reasoning and data understanding, facilitating complex tasks such as question answering [12]. This has been recently confirmed in the visual domain too, by a few preliminary studies that prove *scene graphs* can significantly help VQA [13, 14] as well as image captioning [15] and image retrieval [16].

Nevertheless, one may argue that the migration from structured symbolic representations to black-box distributed ones, is the very reason why deep learning is successful. After all, vector representations can efficiently express an immense space of information while symbolic representations easily become intractable [17]. Therefore, it is essential to make sure any attempt to bring back the advantages of traditional symbolic representations is not a step backwards, and is built upon the success of deep learning.

Inspired by that, we envision a general framework where information extraction tools parse sensory data into a structured semantic embedding that can be expressed symbolically, and higher-level cognitive functions process that graph to perform a downstream task. This general framework could potentially bring many advantages that the conventional end-to-end deep learning lacks. Firstly, information extracted from different data sources or modalities can be represented in the same space, which facilitates multi-modality and multi-source integration. This abstract intermediate representation also means that information extraction and reasoning models can be developed with the flexibility of joint end-to-end training, or training each module separately in the absence of end-to-end supervision. Additionally, extracted knowledge can be easily inspected and interpreted, which makes the pipeline explainable and trustworthy.

Nevertheless, the current state of AI is not capable of extracting such comprehensive, structured, semantic representations from visual and multimedia data, with a practical accuracy and efficiency. In the rest of this chapter, we elaborate the key limitations of present-day AI, and motivate the contributions of this thesis towards addressing those limitations.

1.1 Problem statement

There is a rich, modern literature around extracting semantic information from visual and textual data, which will be studied comprehensively in the next chapter. Here we summarize the main limitations of existing work, and specify the key directions we have pursued to address those limitations.

- Considering the ultimate goal of extracting the comprehensive content of multimedia[1] data within a structured and compact representation, the first requirement is to define a schema[2] for such representations. While the NLP community has studied semantic structures comprehensively, the vision and multimedia research has overlooked the importance of a comprehensive schema. Inspired by recent advances in NLP, we propose Visual Semantic Parsing (Chapter 3) and extend that to Multimedia Event Extraction (Chapter 6), which are both new task formulations that define simple yet powerful representations for visual and multimedia content.

- Once a proper semantic schema is defined, it is essential to develop specialized neural networks to extract such representations from data. Although there are existing models that can extract structured semantics from images[3], they are not able to extract the more advanced and comprehensive structures that we propose in this thesis, and there is little work on extending them to multimedia settings. We propose a collection of new models, such as VSPNet (Chapter 3) and WASE (Chapter 6), which are proven effective for better visual and multimedia understanding.

- One of the biggest limitations of DL models is their need for extensive human supervision, usually in the form of manual annotation on images or text. This severely impedes research

[1] This thesis is focused on images and text, which are considered multimedia when combined.

[2] By schema, we mean an abstract outline of what such semantic representations should look like and what they should have in common. For instance, we argue that any multimedia data should be represented as a graph with entity nodes and predicate nodes, and edges that represent semantic roles.

[3] Those existing methods will be discussed in Chapter 2, and in more detail in Section 3.2.

advancement as it requires significant resources to collect training data needed for experimentation, especially for novel tasks. We develop a collection of weakly supervised learning techniques to train neural networks for complex graph-based tasks, with minimal supervision requirements (Chapters 3 and 6).

- Computer vision models are known for their lack of robustness in the face of physical complexities in visual scenes, such as poor lighting and clutter. It is particularly disappointing that they make nonsensical mistakes that shows their lack of basic common sense knowledge about the world. We present a collection of techniques for reinforcing vision with common sense knowledge, in order to improve their robustness and overall accuracy. We study how to incorporate existing knowledge bases within the process of scene understanding (Chapter 4), as well as how to acquire common sense in a data-driven manner, and how to control the trade-off between constructive common sense versus adverse bias (Chapter 5).

- A prominent limitation of symbolic representations is their closed vocabulary, which reduces their capacity compared to continuous embedding vectors. Due to prohibitive supervision requirements, it is difficult to scale the number of concepts DL models can recognize, which hinders comprehensive information extraction. We have developed a solution to extend our visual understanding technology to open-vocabulary settings using natural, low-cost supervision, which enables AI systems to understand a broader range of semantics and be deployed in more versatile scenarios (Chapter 7).

1.2 Thesis overview

We follow this chapter by a brief literature review around recent advances in computer vision, deep learning, and natural language processing, which are relevant to our goal of structured semantic information extraction from visual and multimedia data. After that, Chapters 3-5 present a collection of methods developed for understanding images via structured representations, including our efforts to acquire and utilize common sense knowledge to reinforce scene understanding.

Chapter 6 attempts to extend our graph-based scene understanding technology to the multimedia domain, by integrating CV methods with NLP counterparts. Furthermore, Chapter 7 presents an extension of our ideas towards open-vocabulary scene understanding, and to make symbolic representations versatile by covering a broad range of semantic concepts. We conclude this thesis in Chapter 8, with remarks on open problems, future research opportunities, and ethical considerations.

Chapter 2: Background and Related Work

Extracting semantic content from raw data is one of the main challenges in AI, and has an immense range of applications. The input data can be in any modality, such as text, image, or video, and the goal is to understand any meaningful information that exists in the data, such as entities, events, relationships, and situations. Computer vision has studied this problem extensively in the context of images and videos, which is usually known as *recognition* and *detection* (of objects [3], actions [18], *etc*.), while NLP has studied this problem in the context of text, under the umbrella terms of *semantic parsing* at the sentence level [19, 20] and *information extraction* at the document level [21, 22]. Despite the distinct literature and terminologies in CV and NLP, many of the concepts are similar, and real-world data is often multimedia, involving both visual and textual content. Therefore, it is essential to study these two problems jointly, and utilize the recent advances in both fields.

For decades, linguists and NLP pioneers have used graph-based, symbolic representations to parse the meaning and knowledge conveyed by language, both to understand languages better, and more recently to extract information from text using AI [11, 19, 20, 21, 22]. The resulting *semantic graphs* or *knowledge graphs* have many applications such as question answering [23, 24] and information retrieval [25, 26]. A common schema for such semantic structures is that some nodes represent what exists (*e.g*. entities, nouns, *etc*.), while some nodes represent what happens (*e.g*. events, verbs, predicates, *etc*.). Other types of node may exist too, along with a variety of edges that create triplet-based facts. An important type of those edges are semantic roles, which represent the role each entity plays in each predicate [11].

Nevertheless, graph-based representations have been less studied in computer vision, perhaps due to the complexities of the (virtually) continuous pixel space compared to the discrete space of words. Similar to NLP, scene understanding aims to detect what exists (objects, scenes, stuff, *etc*.),

and what happens (actions, activities, events, *etc*.) in a given image or video. For consistency, we refer to those groups of concepts as *entities* and *predicates*, respectively. Although there is a rich literature for extracting each of those types of information [3, 18], extracting entities and predicates alone does not result in a full understanding, unless we also extract the relationships between entities and the roles they play in each predicate. Visual Relation Detection (VRD) [27, 28, 29, 30, 31, 32, 33, 34, 35, 36] aims to classify relationships between each pair of detected objects in a scene. These relations include verbs as well as other types of relationship such as comparative and spatial. More recently, Scene Graph Generation (SGG) [37] redefines VRD as a problem of jointly detecting objects and their relationships. This allows semantic reasoning at the image level which results in a better overall performance. Moreover, Human-Object Interaction (HOI) detection [38, 39, 40, 41] is a specialized version of VRD that focuses on verb relations with a human subject.

The formulation of VRD (and hence SGG and HOI) is inherently limiting, as it assumes exactly two roles (arguments) for each predicate: one subject and one object. Hence, VRD methods cannot comprehend many real-world cases where predicates, particularly verbs, have zero or more than one subject/object (*e.g*. `walking` may have no object). Furthermore, some verbs have important arguments other than subject and object. For instance, `racket` is the instrument in `person-hitting-ball` *using* `racket`. Such interactions of more than two entities cannot be expressed as a conventional pairwise relation, and thus are beyond the capability of VRD methods.

Situation Recognition (SR) [42, 43, 44, 45] resolves this limitation, by detecting a verb and all of its arguments in a scene. However, SR assumes there is only one verb in each image, which cannot express concurrent interactions involving various groups of objects. This also means SR methods are designed to extract a simple star-shaped graph with one predicate and a few entities, without the flexibility and complexity of extracting comprehensive semantic graphs that are commonplace in NLP. To address the limitations of both SGG and SR, we have defined Visual Semantic Parsing (VSP) which is a generalized formulation, covering but not limited to SGG and SR. To this end, we represent predicates as nodes (rather than *relation* edges) in the same semantic

space as entity nodes, and instead, represent semantic *roles* (*e.g*. subject, object, instrument, etc.) as edges, resulting in a bipartite graph (Chapter 3).

Semantic graphs extracted from images (*e.g*. by SGG, SR, or VSP) can represent the content of each image in a symbolic space, whose elements are expressed by words. This is potentially a great way to integrate vision with language, since NLP can also extract similar semantic graphs. Nevertheless, there is no research on extracting unified semantic graphs from two modalities jointly. We explore this area in Chapter 6, by extending VSP to multimedia settings, defining the first task that extracts multimedia semantic graphs to understand events.

Extracting graph-based representations is facilitated by the recent advances in graph-structured neural networks [46, 47] (GNN). Many recent CV methods use graph-based message passing to propagate information between region proposals, in order to make context-aware predictions [37, 48, 49, 50]. On the other hand, some methods utilize GNNs on graphs that represent the ontology of concepts, rather than objects in a scene [51, 52, 53, 40]. This often enables generalization to unseen or infrequent concepts by incorporating their relationship with frequently seen concepts. We develop and extend GNNs to adapt them to our new frameworks such as VSP (Chapter 3). We also generalize the two aforementioned ideas to create GNNs that can bridge external knowledge graphs with internal scene graphs, in order to utilize background knowledge in the process of scene understanding (Section 4). Moreover, we develop GNNs to learn *visual commonsense* from scene graphs for the first time, and employ that to enhance the robustness of SGG methods (Section 5).

Another fundamental limitation of SGG is that it requires extensive manual supervision in order to train an accurate model. Specifically, each image in the training data should be annotated by drawing a tight bounding box around each object, labeling each bounding box with a noun, and labeling each pair of objects with a predicate type. Although currently such a dataset is available (Visual Genome [54]), it covers a limited domain with limited types of objects and predicates. Applications that involve other classes would require additional manual labor, which is costly and time-consuming. Considering the fact that bounding box localization is an independent task involving low-level boundary analysis rather than high-level semantic reasoning, we should ideally

disentangle it from SGG, such that localization is learned separately, and SGG does not need localized ground truth graphs for training. Zhang *et al*. [29] recently proposed the only Weakly Supervised VRD method successfully applied on the Visual Genome SGG task. However, the performance is far from fully supervised counterparts. We address this limitation by proposing a new method discussed in Chapter 3, as well as extending to multimedia settings in Chapter 6.

Even if weakly supervised learning becomes possible, extending to more entity and predicate classes would still require extra annotation for every new class, and potentially every new entity/predicate combination. Since the number of possible combinations can quickly become intractable, there is a need for open-vocabulary SGG methods. Few works have aimed at zero-shot object [55] and relation [56, 40] detection. However, this is still an open issue since current zero-shot methods are far from practical performance. In Chapter 7, we explore the idea of naturally supervised learning from image-caption pairs in order to significantly boost the generalization performance of zero-shot models to detect unsupervised concepts.

Chapter 3: Image Understanding Using Semantic Graphs

In this chapter, we introduce the next generation of scene understanding methods, which utilize semantic graphs to extract objects and their interactions from images, by addressing three main limitations in existing Scene Graph Generation (SGG) methods. Firstly, we propose a generalized formulation of SGG, namely Visual Semantic Parsing (VSP), which disentangles entity and predicate recognition, enabling to express more situations through a flexible interaction between entities and predicates. Additionally, we propose the first graph-based weakly supervised learning framework, based on a novel graph alignment algorithm, which enables training without bounding box annotations. Finally, we propose a graph-based neural network architecture named VSPNET, which reduces the computational complexity of mainstream models. Through extensive experiments, we show that VSPNET outperforms weakly supervised baselines significantly and approaches fully supervised performance, while being several times faster. We publicly release the source code of our method[1]. This chapter including all images, figures, tables, equations, and text is based on a recently published collaborative work [57].

3.1 Introduction

The task of Scene Graph Generation (SGG) [37] aims to represent an image with a set of entities (nodes) and predicates (directed edges), as illustrated in Figure 3.1 (bottom). Several methods have been proposed to address this problem [37, 48, 50, 58], but despite their success, important challenges remain unaddressed. Most existing methods are computationally inefficient, as they exhaustively process every pair of object proposals, in order to detect predicates. This results in a quadratic order with respect to the number of proposals. Extending to higher-order interactions has not been studied, and would make this problem even more complex. Furthermore,

[1] https://github.com/alirezazareian/vspnet

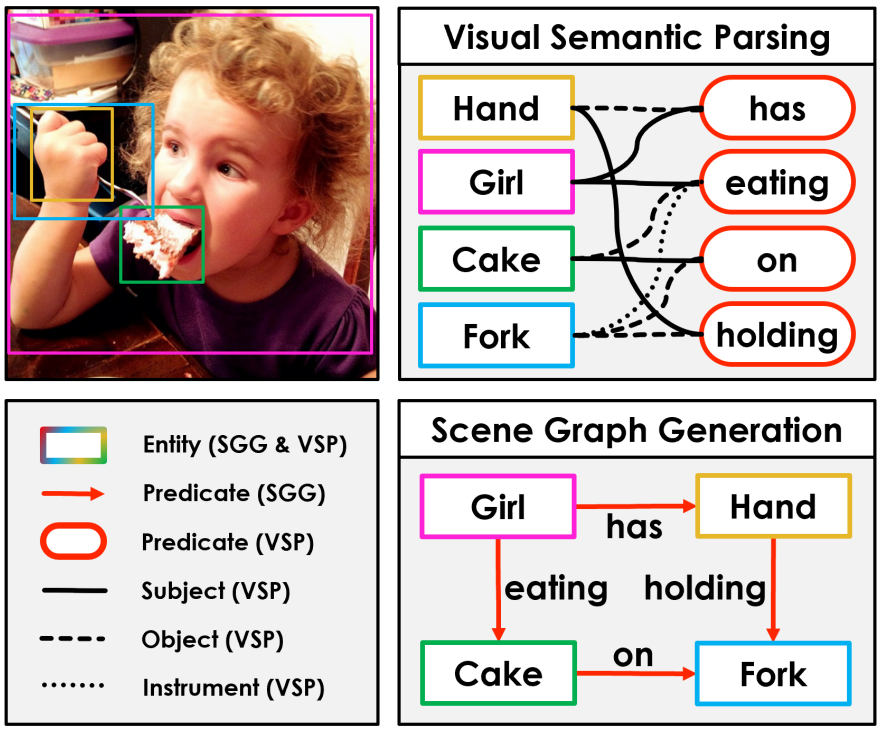


Figure 3.1: An example of structured scene understanding formulated as Scene Graph Generation, where predicates are edges, compared to the proposed Visual Semantic Parsing, where predicates are nodes and edges represent semantic roles.

existing SGG methods require bounding box annotation for each object (node) in ground truth graphs, over the entire training data, which is an expensive constraint. We argue that SGG should ideally be disentangled from bounding box localization, so it can focus on high-level semantic and relational reasoning rather than low-level boundary analysis. However, weakly supervised SGG has barely been studied, and the performance is far from supervised methods [29].

To advance structured scene understanding, we propose the Visual Semantic Parsing Network (VSPNET), which aims to address the three mentioned limitations, *i.e.*, computation and supervision costs, as well as higher-order interactions. To this end, we generalize the formulation of SGG to represent predicates as nodes in the same semantic space as entity nodes, and instead, represent *semantic roles* (*e.g.* subject and object) as edges. Figure 3.1 (top) illustrates the proposed *Visual Semantic Parsing* (VSP) formalism. This not only allows us to break the quadratic complexity, but also can support higher-order interactions that cannot be expressed using the existing SGG formulation. For instance, the semantic structure of a `girl eating cake` *using* `fork` can be

represented as a predicate node, `eating`, connected to three entity nodes `girl`, `cake` and `fork`, via three types of edges that are labeled with *subject*, *object* and *instrument* roles respectively.

Based on this new VSP formulation, we propose a dynamic, attention-based, bipartite message passing framework, which jointly infers node labels and edge labels through an iterative process, resulting in a VSP graph, and in turn a scene graph. VSPNET consists of a *role-driven* attention mechanism to dynamically estimate graph edges, along with a novel three-stage message aggregation network to route messages efficiently throughout the graph. These two modules successively refine nodes and edges of the graph, enabling a joint inference through global reasoning. The proposed architecture does not need to process all pairs of object proposals and hence is computationally efficient. Finally and most importantly, we propose a novel framework to train VSPNET in weakly supervised settings, by defining a two-stage optimization problem and devising a novel graph alignment algorithm to solve it.

Through extensive experiments on the Visual Genome dataset, we show that our method achieves significantly higher accuracy compared to weakly supervised counterparts, approaching fully supervised baselines. We also show that VSPNET is easily extendable to the fully supervised setting, where it can utilize bounding box annotations to further improve performance, and outperform the state of the art. Moreover, we show that our method is several times faster than all baselines, and qualitatively demonstrate its ability to extract higher-order interactions, which are beyond the capability of any existing method.

3.2 Related work

Scene graph generation: The majority of SGG methods start by extracting object proposals from the input image, perform some kind of information propagation (*e.g.* Bi-LSTMs in [58] or Graph Convolutional Nets in [50]) to incorporate context, and then classify each proposal to an entity class, as well as each pair of proposals to a predicate class [37, 48, 58, 49, 59]. This process has a quadratic order and is thus inefficient. Recent methods have tried to reduce the computation by pruning the fully connected graph using a light-weight model [50], or by factorizing the graph

into smaller sub-graphs [49]. However, they still suffer from quadratic order. Newell and Deng [60] proposed a method that does not rely on proposals at all, and directly extracts entities and predicates from a pair of feature maps. Our method is similar in that we allocate a constant, sub-quadratic number of predicates and infer their connection to entities, rather than processing all pairs of entities. In contrast with [60] though, we base our graph on object proposals and exploit message passing to incorporate context.

Neural message passing: Recent deep learning methods have increasingly utilized Message Passing (MP) in various computer vision tasks [61, 62, 40]. Most SGG methods use MP to propagate information among object proposals [37, 48, 49, 50]. Instead of relying on a static, often fully-connected graph, we propose a dynamic, bipartite graph that is refined using attention to route messages between relevant entity-predicate pairs. In contrast with other dynamic MP methods that refine graph edges in each step, which have been used in other tasks such as HOI [41] and video object detection [63], we define edges between entities and predicates rather than pairs of entities, leading to computational efficiency, while incorporating the rich semantic role structure through three-stage aggregation.

Weakly supervised learning: Weak Supervision (WS) has been advocated in several areas, such as object, action, and relation detection [64, 65, 29], and is motivated by the fact that manual annotation of boundaries is time consuming. Most WS object detection methods are based upon multiple instance learning [66], which assumes each ground truth object corresponds to one out of many proposals, but the correspondence is unknown. WSDDN [64] dedicates a network branch to select a proposal for each ground truth. Zhang *et al*. [29] adopted WSDDN for VRD, selecting a pair of proposals for each ground truth relation. In contrast, we define a global optimization problem where the entire output graph has to be aligned with the ground truth graph, rather than considering each predicate independently. Peyre *et al*. [36] defined a global optimization for WS VRD too, but it is limited to a linear regression model for relationship recognition. Our novel WS formulation allows learning with gradient descent, which enables us to train a deep network with a complex message passing architecture.

3.3 Method

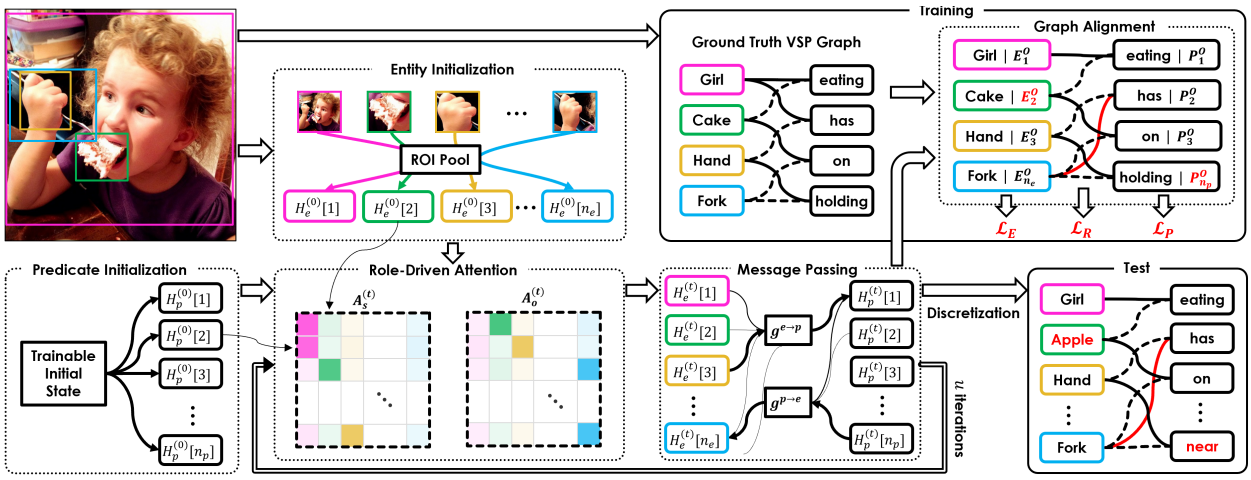


Figure 3.2: Overview of our proposed framework: Given an image, a scene graph is produced by an iterative process involving a multi-headed attention module that infers edges between entities and predicates, and a novel message passing module to propagate information. To define a classification loss for each node and edge, the ground truth graph is aligned to the output graph through a novel weakly supervised algorithm.

In this section, we first formalize our problem in Section 3.3.1, then detail our method and its two-fold contributions: the VSPNET architecture for constructing a semantic graph from an image (Section 3.3.2), and a graph alignment algorithm for weakly supervised training of the proposed network (Section 3.3.3). Figure 3.2 illustrates the general pipeline of our method.

3.3.1 Problem formulation

Given an image I, the goal of SGG is to produce a graph $G_{\text{SGG}} = (\mathcal{N}, \mathcal{E})$ where each node in $\mathcal{N}$ is represented by an entity class $c_i \in C_e$ and a bounding box b_i, and each edge assigns a predicate class to an ordered pair of nodes, *i.e.*, $\mathcal{E} : \mathcal{N} \times \mathcal{N} \mapsto C_p$. The direction of predicate edges usually follow the order they would appear in an English phrase. For instance, a `person sitting on chair` would be represented as an edge labeled `sitting on`, going from the node `person` to the node `chair`, not the other way.

Nevertheless, this notation is inherently limiting, as it restricts predicates to have exactly two

arguments present in the scene. This constraint may be acceptable for relational predicates such as prepositions, but certainly not for verbs, which constitute an important group of predicates. To relax this constraint, we follow [42] to adopt the formulation of Semantic Role Labeling [67], where predicates are represented as nodes, and edges represent semantic roles that entities play in each predicate. Accordingly, we define Visual Semantic Parsing (VSP) as predicting a bipartite graph $G_{\text{VSP}} = (\mathcal{N}_e, \mathcal{N}_p, \mathcal{E})$, where

$$\begin{aligned} \mathcal{N}_e &= \left\{ \left(c_i \in C_e, b_i \in \mathbb{R}^4 \right) \right\}_{i=1}^{n_e}, \\ \mathcal{N}_p &= \left\{ c_k \in C_p \right\}_{k=1}^{n_p}, \text{ and} \\ \mathcal{E} &: \mathcal{N}_p \times \mathcal{N}_e \mapsto C_r. \end{aligned} \tag{3.1}$$

Every scene graph G_{SGG} has an equivalent VSP graph G_{VSP} where each predicate has exactly two roles, subject and object, meaning $C_r = \{s, o\}$. However, an arbitrary VSP graph does not necessarily map to a scene graph, as a predicate may connect to less or more than two entities, potentially involving other semantic roles such as instrument. Hence, VSP is a generalization of SGG.

In this work we employ the VSP formalism, not only because it covers a wider range of semantics, but also because it naturally leads to a more efficient model architecture. In order to consider all possible relationships, most existing methods process a fully connected graph with n_e^2 edges, where n_e is usually the number of proposals which is typically 300. This is while more than 99% of graphs in Visual Genome have less than 20 predicates, and the largest one has 53. VSP allows us to replace the n_e^2 edges with a constant number of predicate nodes n_p, far less than n_e^2.

3.3.2 Visual semantic parsing network

We propose VSPNET, which takes an image as input and generates a VSP graph. To this end, we utilize an object proposal network to initialize a set of *entity nodes*, and devise another module to initialize a set of *predicate nodes*. The goal of VSPNET is to classify each entity and predicate node into entity and predicate classes including background, and classify each entity-predicate pair

into predefined edge types (semantic roles) including no-edge. These are two co-dependent tasks as incorporating nodes would be helpful for edge classification and vice versa. But since both of them are unknown and to be determined, our model successively infers each given the other.

More specifically, VSPNET is based on a novel bipartite message passing framework that propagates information from entities to predicates and vice versa, through a *role-driven* attention mechanism that estimates edges. After nodes are updated using the estimated edges, we update edges by recomputing the attention using the new node representations, and repeat this process for u iterations. To incorporate each semantic role separately, we designate an attention head for each role. This leads to a complex routing problem where messages from a potentially large number of nodes have to be propagated through multiple types of edges. Accordingly, we propose a three-stage message aggregation network to efficiently route and collect relevant messages for updating each node.

Formally, we define $H_e^{(0)} \in \mathbb{R}^{n_e \times d_e}$ to be the initial hidden state of n_e entity nodes, and initialize each row using the appearance (RoI [3]) features of the corresponding object proposal, as well as its bounding box coordinates, by feeding them into two fully connected networks $e_a(.)$ and $e_b(.)$, and adding the two outputs. We also define $H_p^{(0)} \in \mathbb{R}^{n_p \times d_p}$ to be the initial hidden state of n_p predicate nodes. $H_p^{(0)}$ is a trainable matrix, randomly initialized before training but fixed during test. Given $H_e^{(t)}$ and $H_p^{(t)}$, we compute a set of attention matrices $\tilde{A}_r^{(t)} \in \mathbb{R}^{n_p \times n_e}$, each representing a semantic role class r in C_r:

$$\tilde{A}_r^{(t)}[k,i] = \left\langle f_r^p\left(H_p^{(t)}[k]\right) \ , \ f_r^e\left(H_e^{(t)}[i]\right) \right\rangle, \tag{3.2}$$

where $\langle ., . \rangle$ represents dot product, $H[k]$ represents the kth row of H, and f_r^p and f_r^e are trainable fully connected networks to compute the query and key vectors of the attention. We further stack $\tilde{A}_r^{(t)}$ to build the 3-dimensional tensor $\tilde{A}^{(t)}$ that represents the entire role-driven attention. In our experiments, no predicate can take more than one entity for each role, and no entity-predicate pair

can have more than one semantic role. Hence, we normalize $\tilde{A}^{(t)}$ such that:

$$A_r^{(t)}[k,i] = \frac{\exp\left(\tilde{A}_r^{(t)}[k,i]\right)}{p_\varnothing + \sum_{r'=1}^{n_r} \exp\left(\tilde{A}_{r'}^{(t)}[k,i]\right)} \times \frac{\exp\left(\tilde{A}_r^{(t)}[k,i]\right)}{p_\varnothing + \sum_{i'=1}^{n_e} \exp\left(\tilde{A}_r^{(t)}[k,i']\right)}. \tag{3.3}$$

This can be interpreted as applying two softmax functions in paralell on $\tilde{A}^{(t)}$, once normalizing along the axis of roles, and once along the axis of entities, and then multiplying the two normalized matrices, element-wise. The constant $p_\varnothing$ is added to each denominator to allow the sum to be less than one, *e.g.* no role between an entity-predicate pair.

After computing attention matrices, we use them to propagate information from each entity to its relevant predicates and vice versa. To this end, we propose a three-stage message aggregation framework, that computes the incoming message to update each node, by aggregating outgoing messages from all other nodes, and separately processing them in the context of each semantic role. More specifically:

$$\begin{aligned} M_p^{(t)}[k] &= g^{e\to p}\left(A^{(t)}, H_e^{(t)}\right) \\ &= g^{p\leftarrow}\left(\sum_{r=1}^{n_r} g_r^e\left(\sum_{i=1}^{n_e} A_r^{(t)}[k,i] g^{e\to}\left(H_e^{(t)}[i]\right)\right)\right), \end{aligned} \tag{3.4}$$

where $g_r^{e\to}$, g_r^e, and $g_r^{p\leftarrow}$ are independent, trainable fully connected networks, respectively called *send head*, *pool head*, and *receive head*. Note that the pool head consists of n_r separate networks applied on the pooled messages for each role. Similarly, the incoming message to update each entity is computed as:

$$\begin{aligned} M_e^{(t)}[i] &= g^{p\to e}\left(A^{(t)}, H_p^{(t)}\right) \\ &= g^{e\leftarrow}\left(\sum_{r=1}^{n_r} g_r^p\left(\sum_{k=1}^{n_p} A_r^{(t)}[k,i] g^{p\to}\left(H_p^{(t)}[k]\right)\right)\right). \end{aligned} \tag{3.5}$$

After collecting messages for each node, we update their state using two Gated Recurrent Units

(GRU) [68].

$$
\begin{aligned}
H_e^{(t+1)}[i] &= \mathrm{GRU}_e\Big(H_e^{(t)}[i], M_e^{(t)}[i]\Big), \text{ and} \\
H_p^{(t+1)}[k] &= \mathrm{GRU}_p\Big(H_p^{(t)}[k], M_p^{(t)}[k]\Big).
\end{aligned}
\tag{3.6}
$$

This process is repeated for a constant number of times u, and the final states $H_e^{(u)}$ and $H_p^{(u)}$ are passed through another pair of fully connected networks (h_e, h_p) to produce semantic embeddings E^O and P^O for entity and predicate nodes. The final state of the adjacency matrices $A_r^{(u)}$ are stacked together and named A^O.

After the message passing process, we have a continuous and fully differentiable output graph $G^O_{\text{soft}} = (E^O, P^O, A^O)$. In order to produce a valid, discrete graph as defined in Eq. (3.1), we apply a two-step *discretization* process. First, we convert E^O and P^O to discrete labels by picking the nearest neighbor of each of their rows among a dictionary of entity and predicate class embeddings. Next, we threshold the attention matrix A^O and suppress non-maximum roles for each entity-predicate pair. This leads to a discrete graph $G^O = (\mathcal{N}_e^O, \mathcal{N}_p^O, \mathcal{E}^O)$. In the next subsection, we define our cost function, where we also need the opposite process: converting a ground truth graph $G^T = (\mathcal{N}_e^T, \mathcal{N}_p^T, \mathcal{E}^T)$ to a soft representation $G^T_{\text{soft}} = (E^T, P^T, A^T)$. To this end, we stack the class embedding of entity and predicate nodes to get matrices E^T and P^T, and encode the edges into a binary adjacency matrix A^T.

3.3.3 Weakly supervised training

We train our model using pairs of image and unlocalized ground truth graph. Specifically, we need to compare the soft output graph G^O_{soft} (*i.e.* before discretization) to the target G^T_{soft} to calculate a differentiable cost to be minimized. To this end, we find an alignment (*i.e.*, node correspondence) between the two graphs, and then define the overall cost as a summation of loss terms over aligned

nodes and edges. Formally, we define an alignment $\mathcal{I}$ as:

$$\begin{aligned}
&\mathcal{I} = (\mathcal{I}_e, \mathcal{I}_p), \text{ where} \\
&\mathcal{I}_e = \left\{(i,j) | i \in \{1...n_e^O\}, j \in \{1...n_e^T\}\right\}, \text{ and} \\
&\mathcal{I}_p = \left\{(k,l) | k \in \{1...n_p^O\}, l \in \{1...n_p^T\}\right\},
\end{aligned} \tag{3.7}$$

where $n_e^O = n_e$ and $n_p^O = n_p$ are the number of output entity and predicate nodes, while n_e^T and n_p^T are the number of ground truth entity and predicate nodes. $\mathcal{I}_e$ is a valid entity alignment if for any output node i there is at most one target node j, and for each j there is at most one i, where $(i,j) \in \mathcal{I}_e$. A similar constraint holds for $\mathcal{I}_p$. Moreover, $\mathcal{I}_e$ is a *maximal alignment* if all output entities **or** all target entities are aligned, whichever is fewer, *i.e.*

$$\begin{aligned}
&|\mathcal{I}_e| = \min(n_e^O, n_e^T), \text{ and similarly,} \\
&|\mathcal{I}_p| = \min(n_p^O, n_p^T),
\end{aligned} \tag{3.8}$$

where $|.|$ denotes set cardinality. Given an alignment $\mathcal{I}$ between output and target graphs, our objective function is:

$$\mathcal{L}(G^O, G^T, \mathcal{I}) = \mathcal{L}_E + \mathcal{L}_P + \lambda \mathcal{L}_R, \tag{3.9}$$

which is a combination of costs for entity recognition, predicate recognition, and semantic role labeling.

Our weakly supervised training framework is independent of how we define each loss term, as long as they are a summation of costs over aligned nodes. For instance, if we define the entity loss $\mathcal{L}_E$ and predicate loss $\mathcal{L}_P$ as mean square errors of entity and predicate embeddings, and if we define the role loss $\mathcal{L}_R$ to be a binary cross entropy on all attention scores, we can write:

$$\mathcal{L}_E(G^O, G^T, \mathcal{I}) = \frac{1}{|\mathcal{I}_e|} \sum_{(i,j) \in \mathcal{I}_e} \left\| E_i^O - E_j^T \right\|_2^2, \tag{3.10}$$

$$\mathcal{L}_P(G^O, G^T, \mathcal{I}) = \frac{1}{|\mathcal{I}_p|} \sum_{(k,l)\in\mathcal{I}_p} \left\| P_k^O - P_l^T \right\|_2^2, \tag{3.11}$$

$$\mathcal{L}_R(G^O, G^T, \mathcal{I}) = \frac{1}{n_r} \sum_{r=1}^{n_r} \mathcal{L}_r, \tag{3.12}$$

where for role r,

$$\mathcal{L}_r = \frac{1}{|\mathcal{I}|} \sum_{(i,j)\in\mathcal{I}_e} \sum_{(k,l)\in\mathcal{I}_p} \mathcal{X}\big(A_r^O[k,i], A_r^T[l,j]\big), \tag{3.13}$$

where $|\mathcal{I}| = |\mathcal{I}_e||\mathcal{I}_p|$, and

$$\mathcal{X}(p,q) = -q \log p - (1-q)\log(1-p). \tag{3.14}$$

Since $\mathcal{L}_R$ is in a different scale than $\mathcal{L}_E$ and $\mathcal{L}_P$, we use a hyperparameter λ to balance its significance in Eq. (3.9).

The main challenge of weakly supervised learning is that the alignment $\mathcal{I}$ is not known, and thus our training involves the following nested optimization:

$$\phi^* = \arg\min_{\phi} \mathbb{E}\Big[\min_{\mathcal{I}} \mathcal{L}(G^O, G^T, \mathcal{I}) \Big], \tag{3.15}$$

where ϕ is the collection of model parameters that lead to G^O, and the expectation is estimated by averaging over minibatches sampled from training data. Note that the inner optimization is subject to the constraints in Eq. (3.8). Inspired by the EM algorithm [69], we device an alternating optimization approach: We use the Adam Optimizer [70] for the outer optimization, and propose an iterative alignment algorithm to solve the inner optimization in the following.

There are no efficient exact algorithms for solving the inner optimization in Eq. (3.15). Hence, we propose an iterative algorithm to approximate the optimal alignment. We show that given an entity alignment $\mathcal{I}_e$, it is possible to find the optimal predicate alignment $\mathcal{I}_p$ in polynomial time,

and similarly from $\mathcal{I}_p$ to $\mathcal{I}_e$. Accordingly, we perform those two steps iteratively in a coordinate-descent fashion, which is guaranteed to converge to a local optima.

Supposing $\mathcal{I}_e$ is given, we intend to find $\mathcal{I}_p$ that minimizes $\mathcal{L}$. Since $\mathcal{L}_E$ is constant with respect of $\mathcal{I}_p$, the problem reduces to minimizing $\mathcal{L}_P + \lambda\mathcal{L}_R$, which can be written:

$$\mathcal{L}_P + \lambda\mathcal{L}_R = \frac{1}{|\mathcal{I}_p|} \sum_{(k,l)\in\mathcal{I}_p} W^P_{kl}, \tag{3.16}$$

where W^P is a pairwise cost function between output and target predicate nodes, measuring not only their semantic embedding distance, but also the discrepancy of their connectivity in graph. More specifically:

$$\begin{aligned} W^P_{kl} \triangleq & \left\| P^O_k - P^T_l \right\|^2_2 + \\ & \frac{\lambda}{n_r|\mathcal{I}_e|} \sum_{(i,j)\in\mathcal{I}_e} \sum_{r=1}^{n_r} \mathcal{X}\left(A^O_r[k,i], A^T_r[l,j]\right). \end{aligned} \tag{3.17}$$

Note that the optimization of Eq. (3.16) is subject to Eq. (3.8), which makes $|\mathcal{I}_p|$ a constant. Hence, this problem is equivalent to maximum bipartite matching with fully connected cost function W^P, which can be solved in polynomial time using the Kuhn-Munkres algorithm [71].

Similarly, given $\mathcal{I}_p$, we can solve for $\mathcal{I}_e$, and repeat alternation. Every step leads to a lower or equal loss since either $\mathcal{L}_P + \mathcal{L}_R$ is minimized while $\mathcal{L}_E$ is fixed, or $\mathcal{L}_E + \mathcal{L}_R$ is minimized while $\mathcal{L}_P$ is fixed. Since $\mathcal{L}$ cannot become negative, these iterations must converge. We have observed that the convergence value of $\mathcal{L}$ is not sensitive to whether we start by initializing $\mathcal{I}_e$ or $\mathcal{I}_p$, nor does it depend on the initialization value. In our experiments we initialize $\mathcal{I}_p$ to an empty set and proceed with updating $\mathcal{I}_e$. We denote by v the number of iterations used for this alignment procedure.

Our method can be naturally extended to the fully supervised setting by adding a term in Eq. 3.10, to maximize the overlap between the aligned pairs of bounding boxes. Specifically, we

redefine $\mathcal{L}_E$ as:

$$\mathcal{L}_E^{\text{sup}}(G^O, G^T, \mathcal{I}) = \frac{1}{|\mathcal{I}_e|} \sum_{(i,j)\in\mathcal{I}_e} \Big(\Big\| E_i^O - E_j^T \Big\|_2^2 - \lambda_B \log\big(\text{IoU}[B_i^O - B_j^T] + \epsilon\big)\Big), \tag{3.18}$$

where B^O and B^T are the set of output and ground truth bounding boxes respectively, and λ_B and ϵ are hyper-parameters selected by cross-validation. Note that the gradient of the added term with respect to model parameters is zero, and hence this only affects alignment.

3.4 Experiments

We apply our framework on the Visual Genome (VG) dataset [54] for the task of scene graph generation, and compare to both weakly and fully supervised baselines. Through quantitative analysis, we show that VSPNET significantly outperforms the weakly and fully supervised state of the art, while being several times faster than existing methods. Furthermore, ablation experiments show the contribution of each proposed module, namely iterative alignment, role-driven attention, and three-stage message aggregation. We finally provide qualitative evidence that our method is able to produce VSP graphs, which are beyond the expressive capacity of conventional scene graphs.

3.4.1 Implementation details

We use an off-the-shelve Faster R-CNN [3] pretrained on the Open Images dataset [72] to extract object proposals that are needed as inputs to VSPNET. We extract proposal coordinates and features once for all images, and keep them fixed while training and evaluating our model. We do not stack VSPNET on top of Faster R-CNN and do not fine-tune Faster R-CNN during training. We use the original implementation of GRU [68] with 1024-dimensional states (d_e and d_p). The initialization heads e_a and e_b, the attention heads f_r^e and f_r^p, and the message passing heads, $g^{e\rightarrow}$, g_r^e, $g^{p\leftarrow}$, $g^{p\rightarrow}$, g_r^p, and $g^{e\leftarrow}$, are all fully connected networks with two 1024-dimensional layers.

The embedding prediction heads h_e and h_p are each single-layer networks that map 1024-D GRU states to the 300-D embedding space. All fully connected networks use leaky ReLU activation functions [73]. Through cross-validation, we set $\lambda = 10$, $u = 3$, and $v = 3$. We use GloVe embeddings [74] to represent each class, and we fine-tune it during training.

The number of predicate nodes n_p is an important choice. Having more predicate nodes will increase recall but also inference time. Since SGG methods are conventionally evaluated at 100 and 50 predicates, we set $n_p = 100$. To output only 50 predicates, we rank the predicate nodes with respect to their confidence, which is defined as the product of three classification confidence scores, for subject, object and predicate. To report inference time in Table 3.2, we compute the average inference time per image on the test set, using identical settings for all methods (NVIDIA TITAN X, 200 proposals, VGG backbone). The time includes the extraction of proposals and their features.

3.4.2 Task definition

The Visual Genome dataset consists of 108,077 images with manual annotation of objects and relationships, with open-vocabulary classes. [37] and [29] preprocess the annotated objects and relationships to produce scene graphs with a fixed vocabulary. [37] keeps 150 most frequent entity and 50 most frequent predicate classes, while [29] cuts at 200 and 100 respectively. We perform two sets of experiments, based on both [37] and [29], to be able to compare to the performances reported by each paper separately. We follow their preprocessing, data splits, and evaluation protocol, but we assume bounding boxes are not available during weakly supervised training.

The main evaluation metric dubbed SGGEN, measures the accuracy of `subject-predicate-object` triplets. A detected triplet is considered correct if the predicted class for subject, object, and predicate are all correct, and the subject and object bounding boxes have an Intersection over Union (IoU) of at least 0.5 with ground truth. To evaluate, the top K triplets predicted by the model are matched to ground truth triplets. The number of correctly matched triplets is divided by the total number of triplets in the ground truth to compute recall at K. This value is averaged

Table 3.1: Results on VG preprocessed by [29]. All numbers are in percentage and baselines were borrowed from [29]

Method	Supervision	SGGEN		PHRDET	
		R@50	R@100	R@50	R@100
VtransE-MIL [29]	Weak	0.7	0.9	1.5	2.0
PPR-FCN [29]		1.5	1.9	2.4	3.2
VSPNET w/o iterative alignment	Weak	1.3	1.6	8.0	10.2
VSPNET w/ fewer alignment steps		1.8	2.0	9.9	11.9
VSPNET w/o three-stage MP		2.4	2.8	16.7	19.8
VSPNET w/o role-driven MP		2.5	2.9	15.7	18.7
VSPNET w/ fewer MP steps		2.5	2.8	15.5	18.3
VSPNET (Ours)		**3.1**	**3.5**	**17.6**	**20.4**
VtransE [29]	Full	5.5	6.0	9.5	10.4
S-PPR-FCN [29]		6.0	6.9	10.6	11.1
VSPNET (Ours)		**8.9**	**9.9**	**24.0**	**27.8**

over all images leading to R@50 and R@100. Since SGGEN is highly affected by the quality of object proposals. we also report SGCLS, which assumes ground truth bounding boxes are given at test time, instead of proposals. Another metric, PREDCLS assumes ground truth bounding are given, and true object classes are given too. [29] also evaluates using PHRDET, which stands for Phrase Detection. This metric is similar to SGGEN, with the difference that instead of evaluating the bounding box of subject and object separately, the goal is to predict a union bounding box enclosing both the object and subject. To this end, for each detected triplet, we get the union box of its subject and object, and match with that of ground truth triplets at IoU ≥ 0.5.

3.4.3 Results

Table 3.1 shows our quantitative results on VG compared to VtransE [28] and PPR-FCN [29], in both Weakly Supervised (WS) and Fully Supervised (FS) settings, following the evaluation settings of [29]. Our VSPNET achieves the best WS performance, with SGGEN performance more than two times higher and PHRDET more than six times higher than the state of the art. Moreover, the FS extension of our method outperforms the FS variants of those baselines significantly. On the PHRDET measure, even our WS method outperforms all FS baselines. Furthermore, we provide

Table 3.2: Results on VG [37]. Recall numbers (%) are from [50]. Inference time is in seconds per image, partially borrowed from [49].

Method	Supervision	SGGEN			SGCLS		PREDCLS	
		Time	R@50	R@100	R@50	R@100	R@50	R@100
IMP [37]	Full	1.64	3.4	4.2	21.7	24.4	44.7	53.1
MSDN [48]		3.56	7.7	10.5	19.3	21.8	63.1	66.4
MotifNet [58]		2.07	6.9	9.1	23.8	27.2	41.8	48.8
Assoc. Emb. [60]		1.19	9.7	11.3	26.5	30.0	**68.0**	**76.2**
Graph R-CNN [50]		0.83	11.4	13.7	29.6	31.6	54.2	59.1
VSPNET (Ours)		**0.11**	**12.6**	**14.2**	**31.5**	**34.1**	67.4	73.7
VSPNET (Ours)	Weak	**0.11**	**4.7**	**5.4**	**30.5**	**32.7**	**57.7**	**62.4**

ablative variants of our method as extra rows in Table 3.1, to study the effect of each proposed component in isolation.

In VSPNET **w/o iterative alignment**, we replace the proposed alignment algorithm with a heuristic baseline, where we align entities by minimizing $\mathcal{L}_E$ and independently align predicates to minimize $\mathcal{L}_P$, in a one-step process. Our alignment algorithm leads to more than twice the performance of this ablation. We make a similar observation by reducing the number of alignment steps v from 3 to 1, denoted as VSPNET **w/ fewer alignment steps**. Furthermore, in VSPNET **w/o three-stage MP**, we replace the proposed three-stage message aggregation framework with a conventional average pooling, that computes the sum of all messages after multiplying by the attention weights. In VSPNET **w/o role-driven MP**, we keep the three-stage message aggregation, but remove the role-driven attention, and replace $A_r(t)$ with a constant, uniformly distributed attention. Finally, in VSPNET **w/ fewer MP steps**, we only reduce the number of MP steps, u, from 3 to 1. All these three ablations lead to inferior performance, proving the effectiveness of our proposed message passing framework.

To compare to more recent methods, we also perform experiments on the original version of VG that was used by [37], and follow the evaluation protocol of [50]. Table 3.2 compares VSPNET to all the numbers reported by [50]. The FS version of our method outperforms all state-of-the-art methods in all metrics, except slightly outperformed by Assoc. Emb. [60] in PREDCLS only.

In addition to superior accuracy, our method is several times faster than all methods. It is also 5 times faster than Factorizable Net [49], which is the fastest SGG method (0.55 seconds per image), although not shown in Table 3.2, because their reported recall is computed differently than ours.

Furthermore, our WS method shows competitive performance and even outperforms some FS methods. Although there is a performance drop from FS to WS, that is mainly due to the difficulty of object localization in the WS setting. In SGCLS, it achieves a performance very close to FS VSPNET, and outperforms all other FS baselines. This suggests that if some day we have access to very accurate proposals, our WS model would perform as accurately as FS methods. Note that although SGCLS provides ground truth bounding boxes, the WS model only treats them as input proposals, and is still trained with unlocalized ground truth and unknown alignment. Also note that all baselines in Table 3.2 train their Faster R-CNN on VG directly, using annotated bounding boxes that we assume not available in WS settings. Hence, we use an off-the-shelve Faster R-CNN that is pretrained on another dataset in all our experiments. This makes the comparison in Table 3.2 somewhat unfair, to our disadvantage. Adopting the backbone used by the baselines would improve our results, but violates WS constraints.

To illustrate the expressive power of our novel VSP formulation, we train our model on the V-COCO dataset [75], which annotates human actions in images, as well as objects and instruments of those actions. While this dataset has been primarily used for HOI in the literature [41, 76], we adopt it for VSP, by aggregating all action annotations of each image into a single semantic graph, and connecting them to the related objects through 3 types of semantic role: subject, object, and instrument. The resulting VSP graphs have unique properties that are not seen in scene graphs, as shown in Figure 3.3, such as verbs with more than two entities (*e.g.* `person cutting cake` *with* `knife`), and verbs with only one entity (*e.g.* `person smiling`). After training our model on the training set of V-COCO, we apply it on the test set and visualize output graphs in Figure 3.3. Our method successfully generates VSP graphs containing interactions that are not possible with any SGG method.

3.5 Summary

We proposed a method to parse an image into a semantic graph that includes entities, predicates, and semantic roles. Unlike prior works, our method does not require bounding box annotations for training, and does not rely on exhaustive processing of all object proposal pairs. Moreover, it is able to extract more flexible graphs where any number of entities can be involved in each predicate, with a variety of semantic roles. To this end, we proposed a generalized formulation of Scene Graph Generation (SGG) that disentangles predicates from entities, and enables sub-quadratic performance. Based on that, we proposed VSPNET, based on a dynamic, attention-based, bipartite message passing framework. We also introduced the first graph-based weakly supervised learning framework based on a novel graph alignment algorithm. We compared our method to the state of the art through extensive experiments, and achieved significant performance improvements in both weakly supervised and fully supervised settings, while several times faster than every existing method.

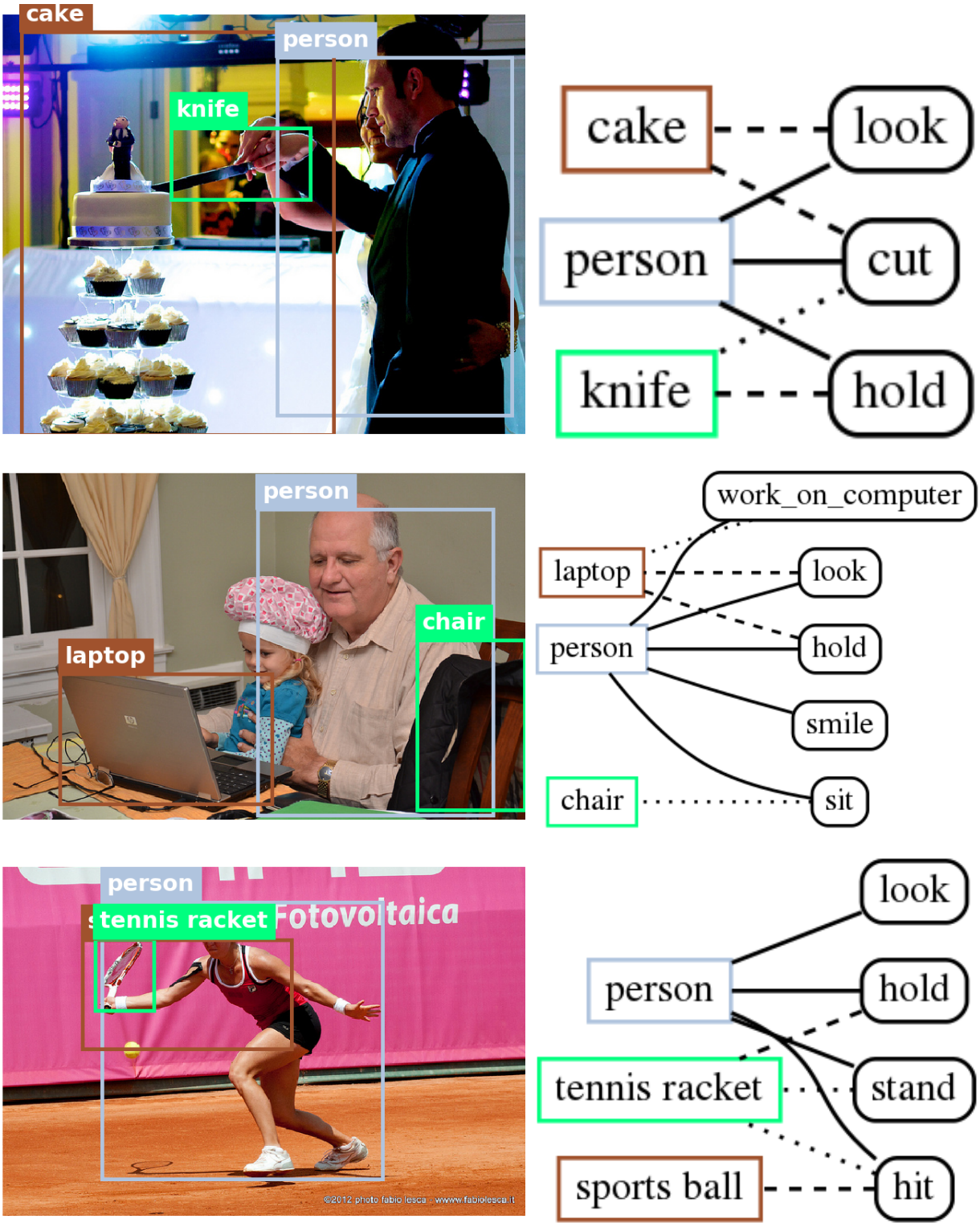


Figure 3.3: Example VSP graphs generated by our method. Solid, dashed, and dotted lines represent subject, object, and instrument.

Chapter 4: Enhancing Scene Graph Generation with External Knowledge Graphs

In the previous chapter, we set the ground for extracting semantic scene graphs from images, and addressed several open issues. Nevertheless, Scene graph generation models are still prone to mistakes due to the challenges of perception in the wild. Perception errors often lead to nonsensical compositions in the output scene graph, which do not follow real-world rules and patterns, and can be corrected using commonsense knowledge. Fortunately, there are rich repositories for commonsense knowledge in form of knowledge graphs, which encode how the world is structured, and how general concepts interact. In this chapter, we utilize external knowledge graphs to improve the quality of scene graphs, by presenting a unified formulation of these two constructs, where a scene graph is seen as an image-conditioned instantiation of a commonsense knowledge graph. Based on this new perspective, we re-formulate scene graph generation as the inference of a bridge between the scene and commonsense graphs, where each entity or predicate instance in the scene graph has to be linked to its corresponding entity or predicate class in the commonsense graph. To this end, we propose a novel graph-based neural network that iteratively propagates information between the two graphs, as well as within each of them, while gradually refining their bridge in each iteration. Our Graph Bridging Network, GB-NET, successively infers edges and nodes, allowing to simultaneously exploit and refine the rich, heterogeneous structure of the interconnected scene and commonsense graphs. Through extensive experimentation, we showcase the superior accuracy of GB-NET compared to the most recent methods, resulting in a new state of the art. We publicly release the source code of our method.[1] This chapter including all images, figures, tables, equations, and text is based on a recently published collaborative work [77].

[1] https://github.com/alirezazareian/gbnet

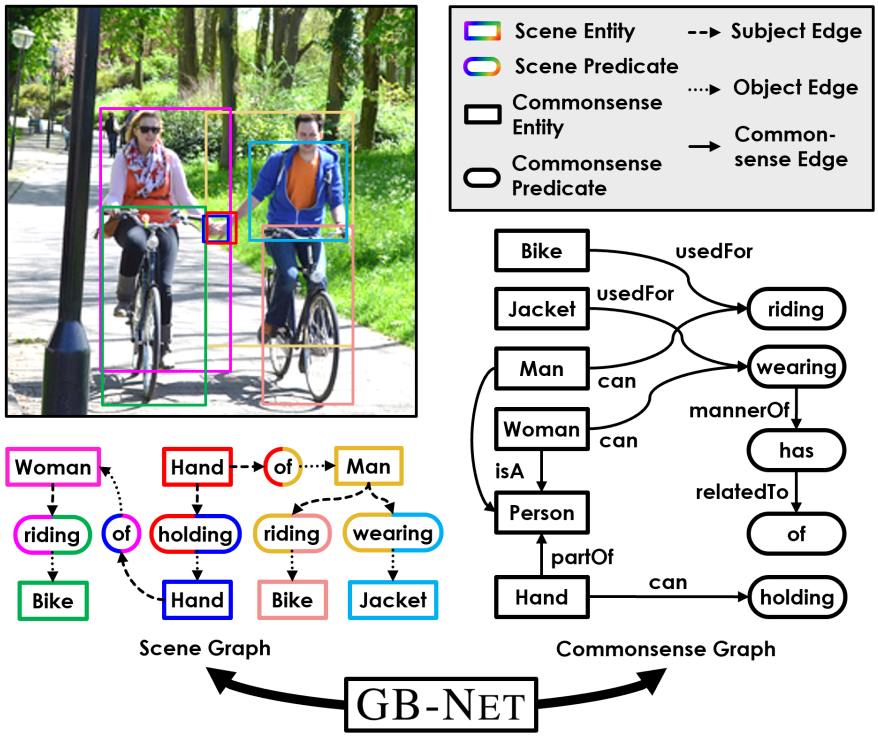


Figure 4.1: Left: An example of a Visual Genome image and its ground truth scene graph. Right: A relevant portion of the commonsense graph. In this chapter we formulate the task of Scene Graph Generation as the problem of creating a bridge between these two graphs. Such bridge not only classifies each scene entity and predicate, but also creates an inter-connected heterogeneous graph whose rich structure is exploited by our method (GB-NET).

4.1 Introduction

Although several SGG methods have been proposed, the state-of-the-art performance for SGG is still far from acceptable. For instance, [78] achieves only 16% mean recall, for matching the top 100 predicted subject-predicate-object triples against ground truth triples. This suggests the current SGG methods are insufficient to address the complexity of this task. Recently, a few papers have attempted to use external *commonsense* knowledge to advance SGG [58, 79, 78], as well as other domains [62, 40]. This commonsense can range from curated knowledge bases such as ConceptNet [80], ontologies such as WordNet [81], or automatically extracted facts such as co-occurance frequencies [58]. The key message of those works is that a prior knowledge about the world can be very helpful when perceiving a complex scene. If we know the relationship

of a `Person` and a `Bike` is most likely `riding`, we can more easily disambiguate between `riding`, `on`, and `attachedTo`, and classify their relationship more accurately. Similarly, if we know a `Man` and a `Woman` are both sub-types of `Person`, even if we only see `Man-riding-Bike` in training data, we can generalize and recognize a `Woman-riding-Bike` triplet at test time. Although this idea is intuitively promising, existing methods that implement it have major limitations, as detailed in Section 4.2, and we address those in the proposed method.

More specifically, recent methods either use ad-hoc heuristics to integrate limited types of commonsense into the scene graph generation process [78, 58], or fail to exploit the rich, graphical structure of commonsense knowledge [79]. To devise a general framework for incorporating any type of graphical knowledge into the process of scene understanding, we take inspiration from early works on knowledge representation and applying structured grammars to computer vision problems [82, 83, 84], and redefine those concepts in the light of the recent advances in graph-based deep learning. Simply put, we formulate both scene and commonsense graphs as knowledge graphs with entity and predicate nodes, and various types of edges. A scene graph node represents an entity or predicate *instance* in a specific image, while a commonsense graph node represents an entity or predicate *class*, which is a general concept independent of the image. Similarly, a scene graph edge indicates the participation of an entity instance (*e.g.* as a subject or object) in a predicate instance in a scene, while a commonsense edge states a general fact about the interaction of two concepts in the world. Figure 4.1 shows an example scene graph and commonsense graph side by side.

Based on this unified perspective, we reformulate the problem of scene graph generation from entity and predicate classification into the novel problem of bridging those two graphs. More specifically, we propose a method that given an image, initializes potential entity and predicate nodes, and then classifies each node by connecting it to its corresponding class node in the commonsense graph, through an edge we call a *bridge*. This establishes a connectivity between instance-level, visual knowledge and generic, commonsense knowledge. To incorporate the rich combination of visual and commonsense information in the SGG process, we propose a novel

graphical neural network, that iteratively propagates messages between the scene and commonsense graphs, as well as within each of them, while gradually refining the bridge in each iteration. Our Graph Bridging Network, GB-NET, successively infers edges and nodes, allowing to simultaneously exploit and refine the rich, heterogeneous structure of the interconnected scene and commonsense graphs.

To evaluate the effectiveness of our method, we conduct extensive experiments on the Visual Genome [54] dataset. The proposed GB-NET outperforms the state of the art consistently in various performance metrics. Through ablative studies, we show how each of the proposed ideas contribute to the results. We also provide further quantitative, qualitative, and speed analysis, to present more insights.

4.2 Related work

As we mentioned before, most SGG methods are based on an object detection backbone that extracts region proposals from the input image. They utilize some kind of information propagation module to incorporate context, and then classify each region to an object class, as well as each pair of regions to a relation class [37, 58, 50, 78]. Our method has two key differences with this conventional process: firstly, our information propagation network operates on a larger graph which consists of not only object nodes, but also predicate nodes and commonsense graph nodes, and has a more complex structure. Secondly, we do not classify each object and relation using classifiers, but instead use a pairwise matching mechanism to connect them to corresponding class nodes in the commonsense graph.

Recently, a few methods [58, 79, 78] have used external knowledge to enhance scene graph generation. This external knowledge is sometimes referred to as "commonsense", because it encodes ontological knowledge about classes, rather than specific instances. Despite encouraging results, these methods have major limitations. Specifically, [58] used triplet frequency to bias the logits of their predicate classifier, and [78] used such frequencies to initialize edge weights on their graphs. Such external priors have been also shown beneficial for recognizing objects [85, 86] and

relationships [87, 88], that are building blocks for SGG. Nevertheless, neither of those methods can incorporate other types or knowledge, such as the semantic hierarchy concepts, or object affordances. Gu *et al.* [79] propose a more general way to incorporate knowledge in SGG, by retrieving a set of relevant facts for each object from a pool of commonsense facts. However, their method does not utilize the structure of the commonsense graph, and treats knowledge as a set of triplets. Our method considers commonsense as a general graph with several types of edges, explicitly integrates that graph with the scene graph by connecting corresponding nodes, and incorporates the rich structure of commonsense by graphical message passing.

Graph-based Neural Networks (GNN) have been used to process external knowledge graphs in various vision tasks [51, 52, 53, 40]. This often enables generalization to unseen or infrequent concepts by incorporating their relationship with frequently seen concepts. This is different from the typical use case of GNNs, which is to contextualize node features in scene graphs [37, 48, 50, 78]. Chen *et al.* [62] were the first to bring those two ideas together, and form a graph by objects in an image as well as object classes in a knowledge graph. Nevertheless, the class nodes in that work were merely an auxiliary means to improve object features before classification. In contrast, we classify the nodes by explicitly inferring their connection to their corresponding class nodes. Moreover, we iteratively refine the bridge between scene and commonsense graphs to enhance our prediction. Furthermore, their task only involves objects and object classes, while we explore a more complex structure where predicates play an important role as well.

4.3 Problem Formulation

In this section, we first formalize the concepts of knowledge graph in general, and commonsense graph and scene graph in particular. Leveraging their similarities, we then reformulate the problem of scene graph generation as bridging these two graphs.

4.3.1 Knowledge graphs

We define a knowledge graph as a set of entity and predicate nodes $(\mathcal{N}_{\mathrm{E}}, \mathcal{N}_{\mathrm{P}})$, each with a semantic label, and a set of directed, weighted edges $\mathcal{E}$ from a predefined set of types. Denoting by Δ a node type (here, either entity E or predicate P), the set of edges encoding the relation r between nodes of type Δ and Δ' is defined as

$$\mathcal{E}_r^{\Delta \to \Delta'} \subseteq \mathcal{N}_\Delta \times \mathcal{N}_{\Delta'} \to \mathbb{R}. \tag{4.1}$$

A commonsense graph is a type of knowledge graph in which each node represents the general concept of its semantic label, and hence each semantic label (entity or predicate class) appears in exactly one node. In such a graph, each edge encodes a relational fact involving a pair of concepts, such as `Hand-partOf-Person` and `Cup-usedFor-Drinking`. Formally, we define the set of commonsense entity (CE) nodes $\mathcal{N}_{\mathrm{CE}}$ and commonsense predicate (CP) nodes $\mathcal{N}_{\mathrm{CP}}$ as all entity and predicate classes in our task. Commonsense edges $\mathcal{E}_{\mathrm{C}}$ consist of 4 distinct subsets, depending on the source and destination node type:

$$\begin{aligned} \mathcal{E}_{\mathrm{C}} = &\{\mathcal{E}_r^{\mathrm{CE}\to\mathrm{CP}}\} \cup \{\mathcal{E}_r^{\mathrm{CP}\to\mathrm{CE}}\} \cup \\ &\{\mathcal{E}_r^{\mathrm{CE}\to\mathrm{CE}}\} \cup \{\mathcal{E}_r^{\mathrm{CP}\to\mathrm{CP}}\}. \end{aligned} \tag{4.2}$$

A scene graph is a different type of knowledge graph where: (a) each scene entity (SE) node is associated with a bounding box, referring to an image region, (b) each scene predicate (SP) node is associated with an ordered pair of SE nodes, namely a subject and an object, and (c) there are two types of undirected edges which connect each SP to its corresponding subject and object respectively. Here because we define knowledge edges to be directed, we model each undirected subject or object edge as two directed edges in the opposite directions, each with a distinct type.

More specifically,

$$\begin{aligned}
\mathcal{N}_{\text{SE}} \subseteq & [0,1]^4 \times \mathcal{N}_{\text{CE}}, \\
\mathcal{N}_{\text{SP}} \subseteq & \mathcal{N}_{\text{SE}} \times \mathcal{N}_{\text{SE}} \times \mathcal{N}_{\text{CP}}, \\
\mathcal{E}_{\text{S}} = & \{\mathcal{E}^{\text{SE}\to\text{SP}}_{\texttt{subjectOf}}, \mathcal{E}^{\text{SE}\to\text{SP}}_{\texttt{objectOf}}, \\
& \mathcal{E}^{\text{SP}\to\text{SE}}_{\texttt{hasSubject}}, \mathcal{E}^{\text{SP}\to\text{SE}}_{\texttt{hasObject}}\},
\end{aligned} \tag{4.3}$$

where $[0,1]^4$ is the set of possible bounding boxes, and $\mathcal{N}_{\text{SE}} \times \mathcal{N}_{\text{SE}} \times \mathcal{N}_{\text{CP}}$ is the set of all possible triples that consist of two scene entity nodes and a scene predicate node. Figure 4.1 shows an example of scene graph and commonsense graph side by side, to make their similarities clearer. Note that our definition of a scene graph can be easily extended to the VSP formulation, by extending the types of edges to include more semantic roles. However, to focus on the Visual Genome evaluation setting, we only study conventional scene graphs in this work.

4.3.2 Bridging knowledge graphs

Considering the similarity between the commonsense and scene graph formulations, we make a subtle refinement in the formulation to bridge these two graphs. Specifically, we remove the class from SE and SP nodes and instead encode it into a set of *bridge* edges $\mathcal{E}_{\text{B}}$ that connect each SE or SP node to its corresponding class, *i.e.*, a CE or CP node respectively:

$$\begin{aligned}
\mathcal{N}^{?}_{\text{SE}} \subseteq & [0,1]^4, \\
\mathcal{N}^{?}_{\text{SP}} \subseteq & \mathcal{N}_{\text{SE}} \times \mathcal{N}_{\text{SE}}, \\
\mathcal{E}_{\text{B}} = & \{\mathcal{E}^{\text{SE}\to\text{CE}}_{\texttt{classifiedTo}}, \mathcal{E}^{\text{SP}\to\text{CP}}_{\texttt{classifiedTo}}, \\
& \mathcal{E}^{\text{CE}\to\text{SE}}_{\texttt{hasInstance}}, \mathcal{E}^{\text{CP}\to\text{SP}}_{\texttt{hasInstance}}\},
\end{aligned} \tag{4.4}$$

where $.^{?}$ means the nodes are implicit, *i.e.*, their classes are unknown. Each edge of type `classifiedTo`, connects an entity or predicate to its corresponding label in the commonsense graph, and has a reverse edge of type `hasInstance` which connects the commonsense node back to the instance. Based on this reformulation, we can define the problem of SGG as the extraction of implicit entity

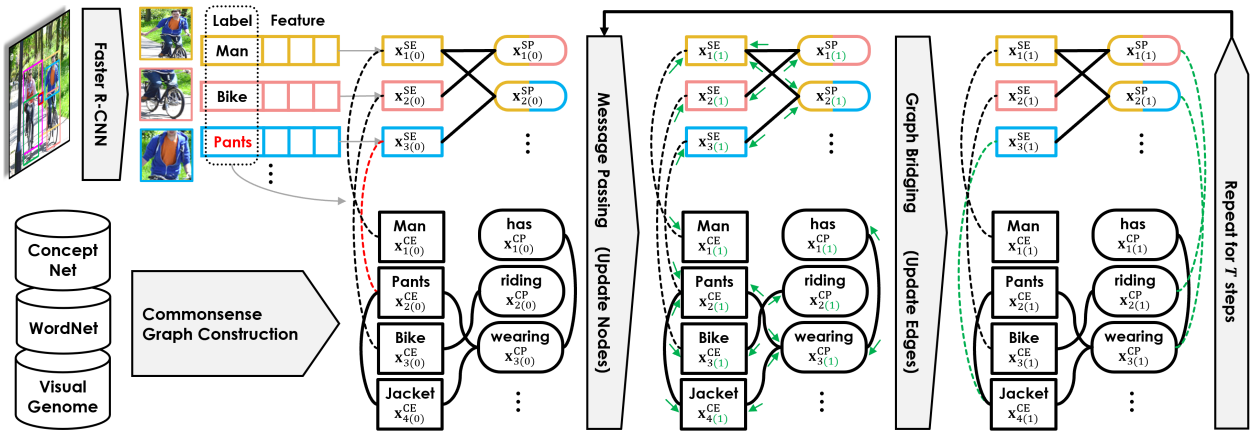


Figure 4.2: An illustrative example of the GB-NET process. First, we initialize the scene graph and entity bridges using a Faster R-CNN. Then we propagate messages to update node representations, and use them to update the entity and predicate bridges. This is repeated T times and the final bridge determines the output label of each node.

and predicate nodes from the image (hereafter called *scene graph proposal*), and then classifying them by connecting each entity or predicate to the corresponding node in the commonsense graph. Accordingly, Given an input image I and a provided and fixed commonsense graph, the goal of SGG with commonsense knowledge is to maximize

$$\begin{aligned} & p(\mathcal{N}_{\mathrm{SE}}, \mathcal{N}_{\mathrm{SP}}, \mathcal{E}_{\mathrm{S}} | I, \mathcal{N}_{\mathrm{CE}}, \mathcal{N}_{\mathrm{CP}}, \mathcal{E}_{\mathrm{C}}) = \\ & \quad p(\mathcal{N}_{\mathrm{SE}}^{?}, \mathcal{N}_{\mathrm{SP}}^{?}, \mathcal{E}_{\mathrm{S}} | I) \times \\ & \quad p(\mathcal{E}_{\mathrm{B}} | I, \mathcal{N}_{\mathrm{CE}}, \mathcal{N}_{\mathrm{CP}}, \mathcal{E}_{\mathrm{C}}, \mathcal{N}_{\mathrm{SE}}^{?}, \mathcal{N}_{\mathrm{SP}}^{?}, \mathcal{E}_{\mathrm{S}}). \end{aligned} \tag{4.5}$$

In this work, the first term is implemented as a region proposal network that infers $\mathcal{N}_{\mathrm{SE}}^{?}$ given the image, followed by a simple predicate proposal algorithm that considers all possible entity pairs as $\mathcal{N}_{\mathrm{SP}}^{?}$. The second term is fulfilled by the proposed GB-NET which infers bridge edges by incorporating the rich structure of the scene and commonsense graphs. Note that unlike most existing methods [58, 78], we do not factorize this into predicting entity classes given the image, and then predicate classes given entities. Therefore, our formulation is more general and allows the proposed method to classify entities and predicates jointly.

4.4 Method

The proposed method is illustrated in Figure 4.2. Given an image, our model first applies a Faster R-CNN [3] to detect objects, and represents them as scene entity (SE) nodes. It also creates a scene predicate (SP) node for each pair of entities, which forms a scene graph proposal, yet to be classified. Given this graph and a background commonsense graph, each with fixed internal connectivity, our goal is to create *bridge* edges between the two graphs that connect each instance (SE and SP node) to its corresponding class (CE and CP node). To this end, our model initializes entity bridges by connecting each SE to the CE that matches the label predicted by Faster R-CNN, and propagates messages among all nodes, through every edge type with dedicated message passing parameters. Given the updated node representations, it computes a pairwise similarity between every SP node and every CP node, and finds maximal similarity pairs to connect scene predicates to their corresponding classes, via predicate bridges. It also does the same for entity nodes to potentially refine their bridges too. Given the new bridges, it propagates messages again, and repeats this process for a predefined number of steps. The final state of the bridge determines which class each node belongs to, resulting in the output scene graph.

4.4.1 Graph initialization

The object detector outputs a set of n detected objects, each with a bounding box b_j, a label distribution p_j and an RoI-aligned [3] feature vector $\mathbf{v}_j$. Then we allocate a *scene entity node* (SE) for each object, and a *scene predicate node* (SP) for each pair of objects, representing the potential predicate with the two entities as its subject and object. Each entity is initialized using its RoI features $\mathbf{v}_j$, and each predicate is initialized using the RoI features $\mathbf{u}_j$ of a bounding box enclosing the union of its subject and object. Formally, we can write, *i.e.*,

$$\mathbf{x}_j^{\mathrm{SE}} = \phi_{\mathrm{init}}^{\mathrm{SE}}(\mathbf{v}_j) \ , \quad \text{and} \quad \mathbf{x}_j^{\mathrm{SP}} = \phi_{\mathrm{init}}^{\mathrm{SP}}(\mathbf{u}_j), \tag{4.6}$$

where $\phi^{\text{SE}}_{\text{init}}$ and $\phi^{\text{SP}}_{\text{init}}$ are two fully connected networks that are branched from the backbone after ROI-align. To form a scene graph proposal, we connect each predicate node to its subject and object via labeled edges. Specifically, we define the following 4 edge types: for a triplet $s - p - o$, we connect p to s using a `hasSubject` edge, p to o using a `hasObject` edge, s to p using a `subjectOf` edge, and o to p using an `objectOf` edge. The reason we have two directions as separate types is that in the message passing phase, the way we use predicate information to update entities should be different from the way we use entities to update predicates.

On the other hand, we initialize the commonsense graph with *commonsense entity nodes* (CE) and *commonsense predicate nodes* (CP) using a linear projection of their word embeddings:

$$\mathbf{x}^{\text{CE}}_i = \phi^{\text{CE}}_{\text{init}}(\mathbf{e}^n_i) \ , \quad \text{and} \qquad \mathbf{x}^{\text{CP}}_i = \phi^{\text{CP}}_{\text{init}}(\mathbf{e}^p_i). \tag{4.7}$$

The commonsense graph also has various types of edges, such as `UsedFor` and `PartOf`, as detailed in Section 4.5.2. Our method is independent of the types of commonsense edges, and can utilize any provided graph from any source.

So far, we have two isolated graphs, scene and commonsense. An SE node representing a detected `Person` intuitively refers to the `Person` concept in the ontology, and hence the `Person` node in the commonsense graph. Therefore, we connect each SE node to the CE node that corresponds the semantic label predicted by Faster R-CNN, via a `classifiedTo` edge type. Instead of a hard classification, we connect each entity to top K_{bridge} classes using p_j (class distribution predicted by Faster R-CNN) as weights. We also create a reverse connection from each CE node to corresponding SE nodes, using an `hasInstance` edge, but with the same weights p_j. As mentioned earlier, this is to make sure information flows from commonsense to scene as well as scene to commonsense, but not in the same way. We similarly define two other edge types, `classifiedTo` and `hasInstance` for predicates, which are initially an empty set, and will be updated to bridge SP nodes to CP nodes as we explain in the following. These 4 edge types can be seen as flexible *bridges* that connect the two fixed graphs, which are considered latent variables to

be determined by the model.

This forms a heterogeneous graph with four types of nodes (SE, SP, CE, and CP) and various types of edges: scene graph edges $\mathcal{E}_{\mathrm{S}}$ such as `subjectOf`, commonsense edges $\mathcal{E}_{\mathrm{C}}$ such as `usedFor`, and bridge edges $\mathcal{E}_{\mathrm{B}}$ such as `classifiedTo`. Next, we explain how our proposed method updates node representations and bridge edges, while keeps commonsense and scene edges constant.

4.4.2 Successive message passing and bridging

Given a heterogeneous graph as described above, we employ a variant of GGNN [89] to propagate information among nodes. First, each node representation is fed into a fully connected network to compute *outgoing* messages, that is

$$\mathbf{m}_i^{\Delta\rightarrow} = \phi_{\mathrm{send}}^{\Delta}(\mathbf{x}_i^{\Delta}), \tag{4.8}$$

for each i and node type Δ, where ϕ_{send} is a trainable *send head* which has shared weights across nodes of each type. After computing outgoing messages, we send them through all outgoing edges, multiplying by the edge weight. Then for each node, we aggregate incoming messages, by first adding across edges of the same type, and then concatenating across edge types. We compute the *incoming* message for each node by applying another fully connected network on the aggregated messages:

$$\mathbf{m}_j^{\Delta\leftarrow} = \phi_{\mathrm{receive}}^{\Delta}\left(\bigcup_{\Delta'}\bigcup^{\mathcal{E}_k\in\mathcal{E}^{\Delta'\rightarrow\Delta}}\sum_{(i,j,a_{ij}^k)\in\mathcal{E}_k} a_{ij}^k\mathbf{m}_i^{\Delta'\rightarrow}\right), \tag{4.9}$$

where ϕ_{receive} is a trainable *receive head* and $\cup$ denotes concatenation. Note that the first concatenation is over all 4 node types, the second concatenation is over all edge types from Δ' to Δ, and the sum is over all edges of that type, where i and j are the head and tail nodes, and a_{ij}^k is the edge weight. Given the incoming message for each node, we update the representation of the node

using a Gated Recurrent Unit (GRU) update rule, following [68]:

$$\begin{aligned}
\mathbf{z}_j^{\Delta} &= \sigma\big(W_z^{\Delta}\mathbf{m}_j^{\Delta\leftarrow} + U_z^{\Delta}\mathbf{x}_j^{\Delta}\big),\\
\mathbf{r}_j^{\Delta} &= \sigma\big(W_r^{\Delta}\mathbf{m}_j^{\Delta\leftarrow} + U_r^{\Delta}\mathbf{x}_j^{\Delta}\big),\\
\mathbf{h}_j^{\Delta} &= \tanh\big(W_h^{\Delta}\mathbf{m}_j^{\Delta\leftarrow} + U_h^{\Delta}(\mathbf{r}_j^{\Delta}\odot\mathbf{x}_j^{\Delta})\big),\\
\mathbf{x}_j^{\Delta} &\Leftarrow (1-\mathbf{z}_j^{\Delta})\odot\mathbf{x}_j^{\Delta} + \mathbf{z}_j^{\Delta}\odot\mathbf{h}_j^{\Delta},
\end{aligned} \tag{4.10}$$

where σ is the sigmoid function, and $W_{\cdot}^{\Delta}$ and $U_{\cdot}^{\Delta}$ are trainable matrices that are shared across nodes of the same type, but distinct for each node type Δ. This update rule can be seen as an extension of GGNN [89] to heterogeneous graphs, with a more complex message aggregation strategy. Note that $\Leftarrow$ means we update the node representation. Mathematically, this means $\mathbf{x}_{j(t+1)}^{\Delta} = U(\mathbf{x}_{j(t)}^{\Delta})$, where U is the aforementioned update rule and (t) denotes iteration number. For simplicity, we drop this subscript throughout this chapter.

So far, we have explained how to update node representations using graph edges. Now using the new node representations, we should update the bridge edges $\mathcal{E}_{\mathrm{B}}$ that connect scene nodes to commonsense nodes. To this end, we compute a pairwise similarity from each SE to all CE nodes, and from each SP to all CP nodes.

$$\mathbf{a}_{ij}^{\mathrm{EB}} = \frac{\exp\langle \mathbf{x}_i^{\mathrm{SE}}, \mathbf{x}_j^{\mathrm{CE}}\rangle_{\mathrm{EB}}}{\sum_{j'}\exp\langle \mathbf{x}_i^{\mathrm{SE}}, \mathbf{x}_{j'}^{\mathrm{CE}}\rangle_{\mathrm{EB}}}, \quad \text{where} \quad \langle \mathbf{x}, \mathbf{y}\rangle_{\mathrm{EB}} = \phi_{\mathrm{att}}^{\mathrm{SE}}(\mathbf{x})^T\phi_{\mathrm{att}}^{\mathrm{CE}}(\mathbf{y}), \tag{4.11}$$

and similarly for predicates,

$$\mathbf{a}_{ij}^{\mathrm{PB}} = \frac{\exp\langle \mathbf{x}_i^{\mathrm{SP}}, \mathbf{x}_j^{\mathrm{CP}}\rangle_{\mathrm{PB}}}{\sum_{j'}\exp\langle \mathbf{x}_i^{\mathrm{SP}}, \mathbf{x}_{j'}^{\mathrm{CP}}\rangle_{\mathrm{PB}}}, \quad \text{where} \quad \langle \mathbf{x}, \mathbf{y}\rangle_{\mathrm{PB}} = \phi_{\mathrm{att}}^{\mathrm{SP}}(\mathbf{x})^T\phi_{\mathrm{att}}^{\mathrm{CP}}(\mathbf{y}). \tag{4.12}$$

Here $\phi_{\mathrm{att}}^{\Delta}$ is a fully connected network that resembles *attention head* in transformers. Note that since $\phi_{\mathrm{att}}^{\Delta}$ is not shared across node types, our similarity metric is asymmetric. We use each $\mathbf{a}_{ij}^{\mathrm{EB}}$ to set the edge weight of the `classifiedTo` edge from $\mathbf{x}_i^{\mathrm{SE}}$ to $\mathbf{x}_j^{\mathrm{CE}}$, as well as the `hasInstance` edge from $\mathbf{x}_j^{\mathrm{CE}}$ to $\mathbf{x}_i^{\mathrm{SE}}$. Similarly we use each $\mathbf{a}_{ij}^{\mathrm{PB}}$ to set the weight of edges between $\mathbf{x}_i^{\mathrm{SP}}$ and

$\mathbf{x}_j^{\text{CP}}$. In preliminary experiments we realised that such fully connected bridges hurt performance in large graphs. Hence, we only keep the top K_{bridge} values of $\mathbf{a}_{ij}^{\text{EB}}$ for each i, and set the rest to zero. We do the same thing for predicates, keeping the top K_{bridge} values of $\mathbf{a}_{ij}^{\text{PB}}$ for each i. Given the updated bridges, we propagate messages again to update node representations, and iterate for a fixed number of steps, T. The final values of $\mathbf{a}_{ij}^{\text{EB}}$ and $\mathbf{a}_{ij}^{\text{PB}}$ are the outputs of our model, which can be used to classify each entity and predicate in the scene graph.

4.4.3 Training

We closely follow [78] which itself follows [58] for training procedure. Specifically, given the output and ground truth graphs, we align output entities and predicates to ground truth counterparts. To align entities we use IoU and predicates will be aligned naturally since they correspond to aligned pairs of entities. Then we use the output probability scores of each node to define a cross-entropy loss. The sum of all node-level loss values will be the objective function to be minimized using Adam [70].

Due to the highly imbalanced predicate statistics in Visual Genome, we observed that best-performing models usually concentrate their performance merely on the most frequent classes such as `on` and `wearing`. To alleviate this, we modify the basic cross-entropy objective that is commonly used by assigning an importance weight to each class. We follow the recently proposed class-balanced loss [90] where the weight of each class is inversely proportional to its frequency. More specifically, we use the following loss function for each predicate node:

$$\mathcal{L}_i^P = -\frac{1-\beta}{1-\beta^{n_j}} \log \mathbf{a}_{ij}^{\text{PB}}, \tag{4.13}$$

where j is the class index of the ground truth predicate aligned with i, n_j is the frequency of class j in training data, and β is a hyperparameter. Note that $\beta = 0$ leads to a regular cross-entropy loss, and the more it approaches 1, the more strictly it suppresses frequent classes. To be fair in comparison with other methods, we include a variant of our method without reweighting, which

still outperforms all other methods.

4.5 Experiments

Following the literature, we use the large-scale Visual Genome benchmark [54] to evaluate our method. We first show our GB-NET outperforms the state of the art, by extensively evaluating it on 24 performance metrics. Then we present an ablation study to illustrate how each innovation contributes to the performance. We also provide a per-class performance breakdown to show the consistency and robustness of our performance across frequent and rare classes. That is accompanied by a computational speed analysis, and several qualitative examples of our generated graphs compared to the state of the art, side by side.

4.5.1 Task description

Visual Genome [54] consists of 108,077 images with annotated objects (entities) and pairwise relationships (predicates), which is then post-processed by [37] to create scene graphs. They use the most frequent 150 entity classes and 50 predicate classes to filter the annotations. Figure 4.1 shows an example of their post-processed scene graphs which we use as ground truth. We closely follow their evaluation settings such as train and test splits.

The task of scene graph generation, as described in Section 4.4, is equivalent to the SGGEN scenario proposed by [37] and followed ever since. Given an image, the task of SGGEN is to jointly infer entities and predicates from scratch. Since this task is limited by the quality of the object proposals, [37] also introduced two other tasks that more clearly evaluate entity and predicate recognition. In SGCLS, we take localization (here region proposal network) out of the picture, by providing the model with ground truth bounding boxes during test, simulating a *perfect* proposal model. In PREDCLS, we take object detection for granted, and provide the model with not only ground truth bounding boxes, but also their true entity class. In each task, the main evaluation metric is average per-image recall of the top K subject-predicate-object triplets. The confidence of a triplet that is used for ranking is computed by multiplying the classification confidence of all

three elements. Given the ground truth scene graph, each predicate forms a triplet, which we match against the top K triplets in the output scene graph. A triplet is matched if all three elements are classified correctly, and the bounding boxes of subject and object match with an IoU of at least 0.5. Besides the choice of K, there are two other choices to be made: (1) Whether or not to enforce the so-called *Graph Constraint* (GC), which limits the top K triplets to only one predicate for each ordered entity pair, and (2) Whether to compute the recall for each predicate class separately and take the mean (mR), or compute a single recall for all triplets (R) [78]. We comprehensively report both mean and overall recall, both with and without GC, and conventionally use both 50 and 100 for K, resulting in 8 metrics for each task, 24 in total.

4.5.2 Implementation details

We use three-layer fully connected networks with ReLU activation for all trainable networks ϕ_{init}, ϕ_{send}, ϕ_{receive} and ϕ_{att}. We set the dimension of node representations to 1024, and perform 3 message passing steps, except in ablation experiments where we try 1, 2 and 3. We tried various values for β. Generally the higher it is, mean recall improves and recall falls. We found 0.999 is a good trade-off, and chose $K_{\text{bridge}} = 5$ empirically. All hyperparameters are tuned using a validation set randomly selected from training data. We borrow the Faster R-CNN trained by [58] and shared among all our baselines, which has a VGG-16 backbone and predicts 128 proposals.

In our commonsense graph, the nodes are the 151 entity classes and 51 predicate classes that are fixed by [37], including background. We use the GloVE [74] embedding of category titles to initialize their node representation (via ϕ_{init}), and fix GloVE during training. For categories with a title longer than one word, we use an average of the GloVE embedding of each word. We compile our commonsense edges from three sources, WordNet [81], ConceptNet [80], and Visual Genome. To summarize, there are three groups of edge types in our commonsense graph. We have `SimilarTo` from WordNet hierarchy, we have `PartOf`, `RelatedTo`, `IsA`, `MannerOf`, and `UsedFor` from ConceptNet, and finally from VG training data we have conditional probabilities of subject given predicate, predicate given subject, subject given object, *etc.* Inspired in part by

[40], we use three similarity metrics of the WordNet API (namely path similarity, LCH similarity, and WUP similarity) to determine whether two entity classes are relevant or not. This is encoded in the edge type `WordNetSimilarTo`. This strategy does not work well for predicate classes, so this edge is only between pairs of entities. From ConceptNet, we choose five relationships that frequently exist between our nodes, namely `PartOf`, `RelatedTo`, `IsA`, `MannerOf`, and `UsedFor`. As discussed earlier in Section 4.4, for directed relationships such as `PartOf`, we create a reverse edge type (in this case `HasPart`) too, which has the same adjacency matrix as `PartOf`, transposed. Finally, we use Visual Genome to get co-occurrence statistics between categories, inspired by [58] but in a more comprehensive manner. We estimate conditional probabilities of subject given predicate, object given predicate, predicate given subject, predicate given object, subject given object, and object given subject, as well as the covariance of entity classes as they connect to the same predicate, and the covariance of predicate classes as they connect to the same entity. These edge types capture a variety of statistical interactions between classes. Overall these three sources lead to 19 edge types (including backward edge types for asymmetric relationships). The process of compiling and pruning the knowledge graph is semi-automatic and takes less than a day from a single person. We make it publicly available as a part of our code. We have also tried using each individual source (e.g. only ConceptNet) independently, which requires less effort, and does not significantly impact the performance. There are also recent approaches to automate the process of commonsense knowledge graph construction [91, 92], which can be utilized to further reduce the manual labor.

4.5.3 Main results

Table 4.1 summarizes our results in comparison to the state of the art. IMP+ refers to the re-implementation of [37] by [58] using their new Faster R-CNN backbone. That method does not use any external knowledge and only uses message passing among the entities and predicates and then classifies each. Hence, it can be seen as a strong, but knowledge-free baseline. FREQ is a simple baseline proposed by [58], which predicts the most frequent predicate for any given pair

of entity classes, solely based on statistics from the training data. FREQ surprisingly outperforms IMP+, confirming the efficacy of commonsense in SGG.

SMN [58] applies bi-directional LSTMs on top of the entity features, then classifies each entity and each pair. They bias their classifier logits using statistics from FREQ, which improves their total recall significantly, at the expense of higher bias against less frequent classes, as revealed by [78]. More recently, KERN [78] encodes VG statistics into the edge weights of the graph, which is then incorporated by propagating messages. Since it encodes statistics more implicitly, KERN is less biased compared to SMN, which improves mR. Our method improves both R and mR significantly, and our class-balanced model, GB-NET-β, further enhances mR (+2.7% in average) without hurting R by much (−0.2%).

We observed that the state of the art performance has been saturated in the SGGEN setting, especially for overall recall. This is partly because object detection performance is a bottleneck that limits the performance. It is worth noting that mean recall is a more important metric than overall recall, since most SGG methods tend to score a high overall recall by investing on few most frequent classes, and ignoring the rest [78]. As shown in Table 4.1, our method achieves significant improvements in mean recall.

4.5.4 Ablation study

To further explain our performance improvement, Table 4.2 compares our full method with its weaker variants. Specifically, to investigate the effectiveness of commonsense knowledge, we remove the commonsense graph and instead classify each node in our graph using a 2-layer fully connected classifier after message passing. This negatively impacts performance in all metrics, proving our method is able to exploit commonsense knowledge through the proposed bridging technique. Moreover, to highlight the importance of our proposed message passing and bridge refinement process, we repeated the experiments with fewer steps. We observe the performance drops significantly with fewer steps, proving the effectiveness of our model, but it saturates as we go beyond 3 steps.

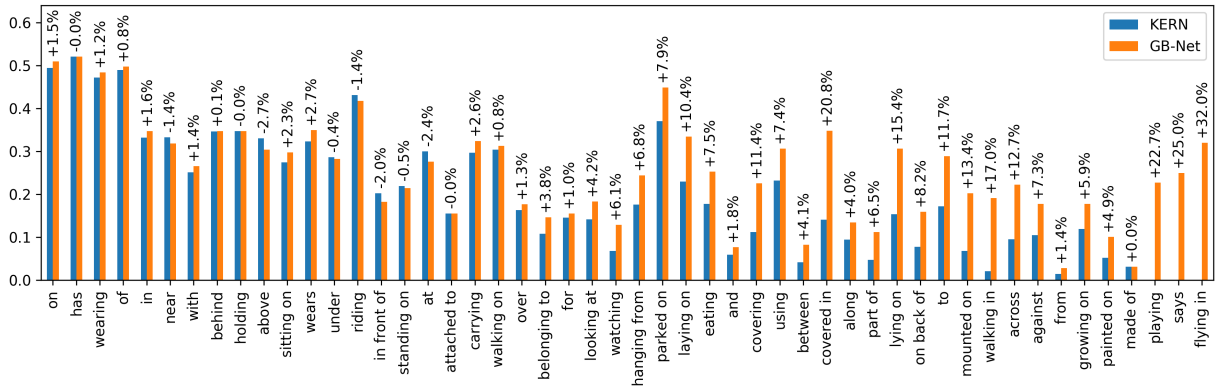


Figure 4.3: Comparison of our method GB-NET with KERN [78] in terms of recall at 50 per predicate class, without graph constraint. The horizontal axis was ordered decreasingly based on frequency in VG.

4.5.5 Per-class performance

Figure 4.3 illustrates the recall of our method for each predicate class separately, where predicates are ordered decreasingly according to their frequency in the training set. While state-of-the-art methods such as KERN [78] obtain much lower performance on the tail of the distribution, our method significantly improves the performance of the tail without losing on the frequent predicates, resulting in a more reliable and consistent performance overall.

4.6 Computational cost

We compute the training and test speed of our method and compare to KERN [78] using identical hardware, with one GPU of type NVIDIA GeForce GTX 1080 Ti with 11 gigabytes of memory, and summarize the results in Table 4.3. Perhaps the most important aspect of computation is the run time when deploying the model on new images. To this end, we run each trained model on the entire test set of Visual Genome (VG), *i.e.* 26446 images, and get the average run time over all images in terms of seconds. Our method is 34% faster than the state of the art, while being significantly more accurate as demonstrated earlier.

Another important factor is the duration of training. We record the time it takes to train each

model on one epoch of the VG training set, *i.e.* 56224 images, and get the average over 10 training epochs. As Table 4.3 shows, our method is more than twice faster than the state of the art. One of the reasons is that KERN has two stages of message passing, each with three steps, first to infer entities, and then to infer predicates, while our method infers both entities and predicates jointly, through 3 steps of global message passing.

Finally, we compare the number of trainable parameters each method has. Our method has 10% more parameters than KERN, while it is 52% and 34% faster than KERN during training and test respectively. Note that in all methods, 139.8 millions of the parameters belong to the Faster R-CNN detector, which we fix while training for scene graph generation.

4.7 Qualitative results

To demonstrate the performance of our method qualitatively, Figures 4.4-4.16 show examples of scene graphs generated by our method, compared to the ones generated by KERN. These examples illustrate how our method predicts more commonsensical graphs despite visual ambiguities in the scene. We observed several patterns in which our method outperforms KERN. Since KERN (and most other SGG methods such as [58]) first classify each entity and then classify predicates, they are unable to utilize predicate semantics to enhance entity classification. Thus in many cases, KERN misclassifies an entity, due to visual ambiguity and clutter, while our method makes the correct prediction that might be less apparent visually, but lead to a more consistent scene graph semantically. In some other cases, KERN misclassifies an entity not because it is visually cluttered, but because the bounding box is too loose and covers a big portion of background. Our method is more robust to such bounding box inaccuracies, resulting a higher overall performance. Finally, in some cases entities are classified correctly by KERN, but the choice of predicates is inappropriate. Our method usually picks the correct predicate, in accordance to commonsense knowledge, such as object affordances. More detailed discussion can be found in each figure's caption.

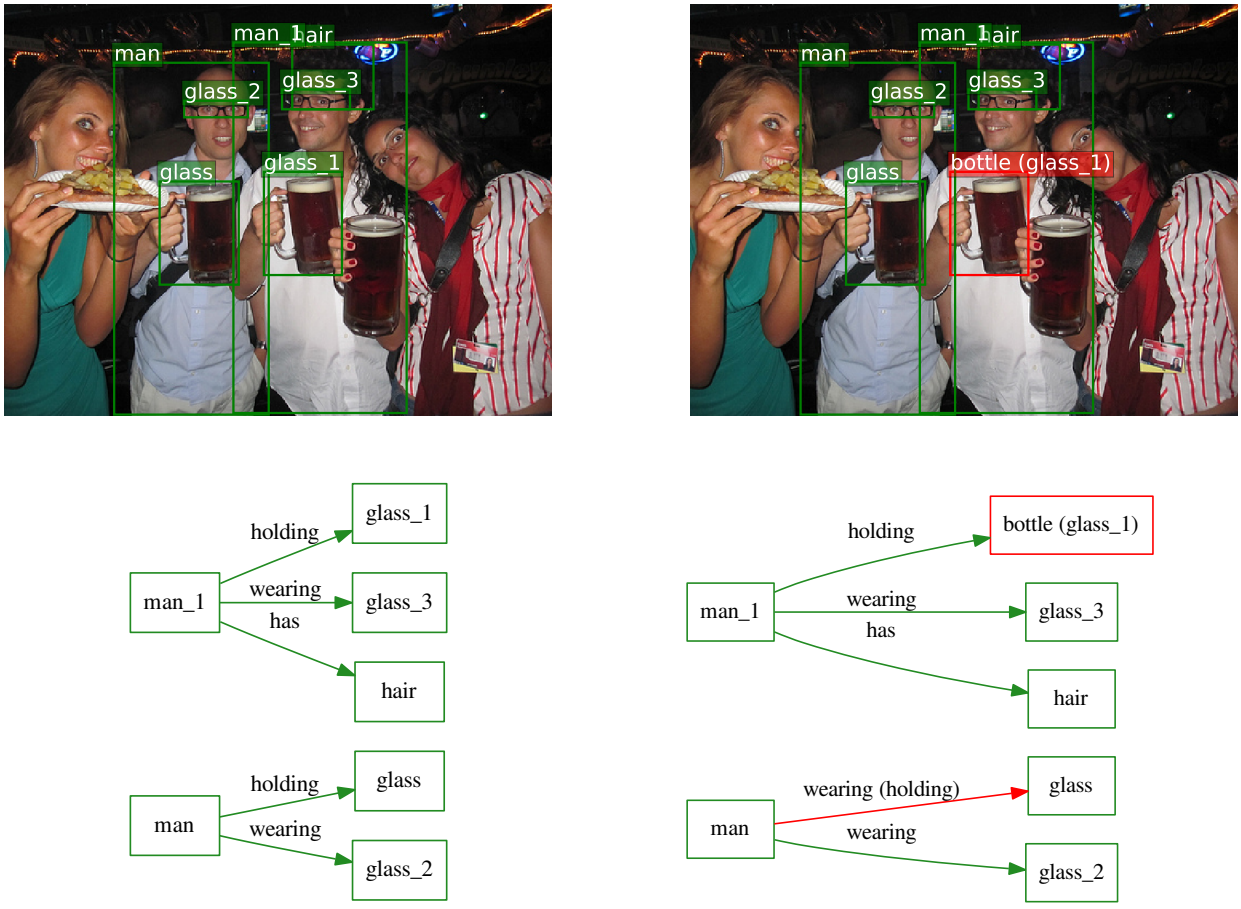


Figure 4.4: Example comparison of our method GB-NET (left) with KERN [78] (right). Misclassified entities and predicates are colored red, and the correct class is included in parentheses. This is a challenging image with 4 occurrences of "glass" with two different meanings (eyeglasses and beer glass). Our method is able to choose the appropriate relation (wearing or holding) for each instance. KERN mistakes a glass for a bottle and predicts a "wearing" relation between a man and his drink.

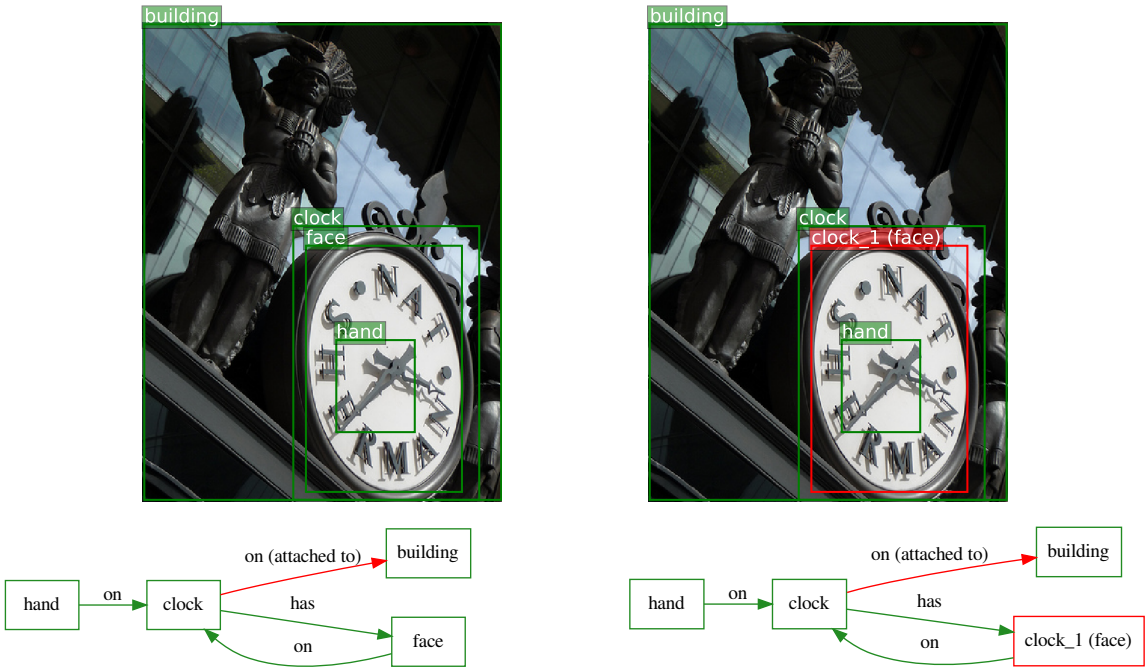


Figure 4.5: Example comparison of our method GB-NET (left) with KERN [78] (right). The concept of a clock face is challenging for KERN but our method can produce such output, by exploiting the prior knowledge and statistics that clocks can have faces and the face would be on the clock. KERN predicts the triplet clock has clock, which does not make sense.

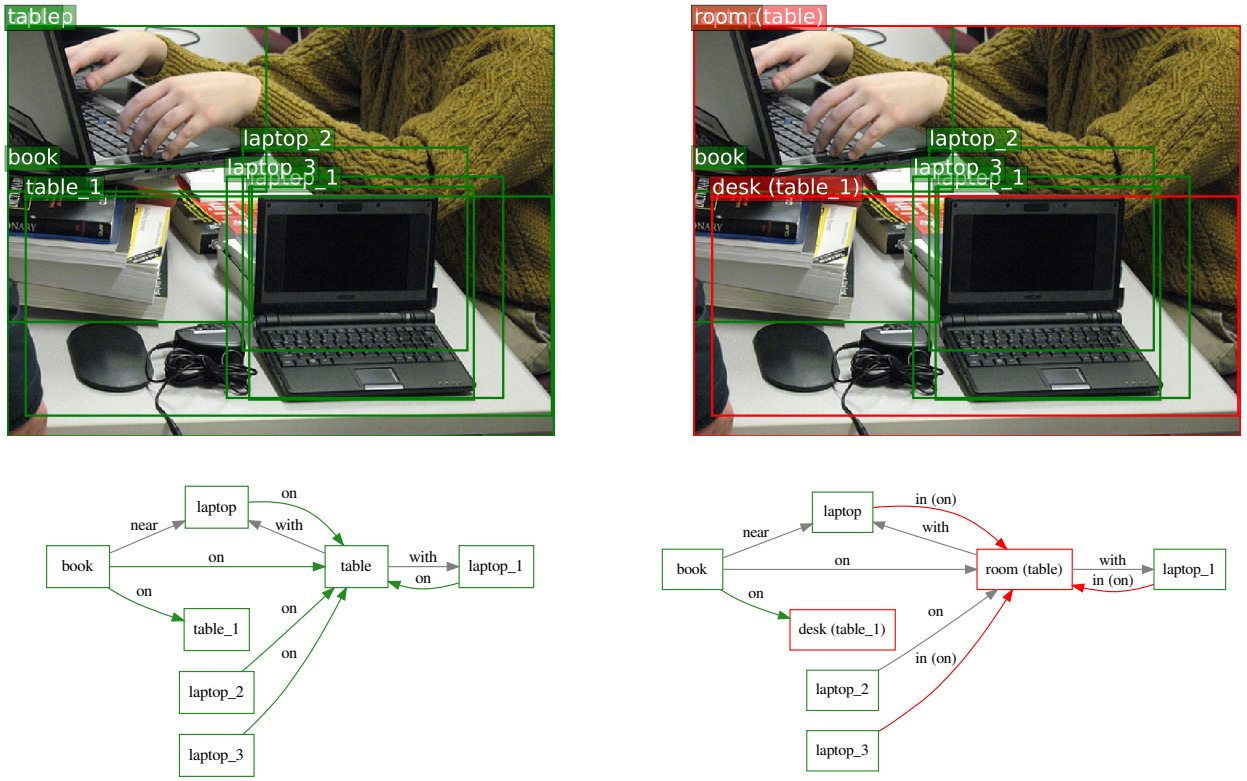


Figure 4.6: Example comparison of our method GB-NET (left) with KERN [78] (right). KERN misclassifies the table as a room, possibly because the bounding box contains the entire scene, but this leads to incorrect triplets such as laptop on room. Our method predicts the more appropriate class table, that makes every triplet more commonsensical.

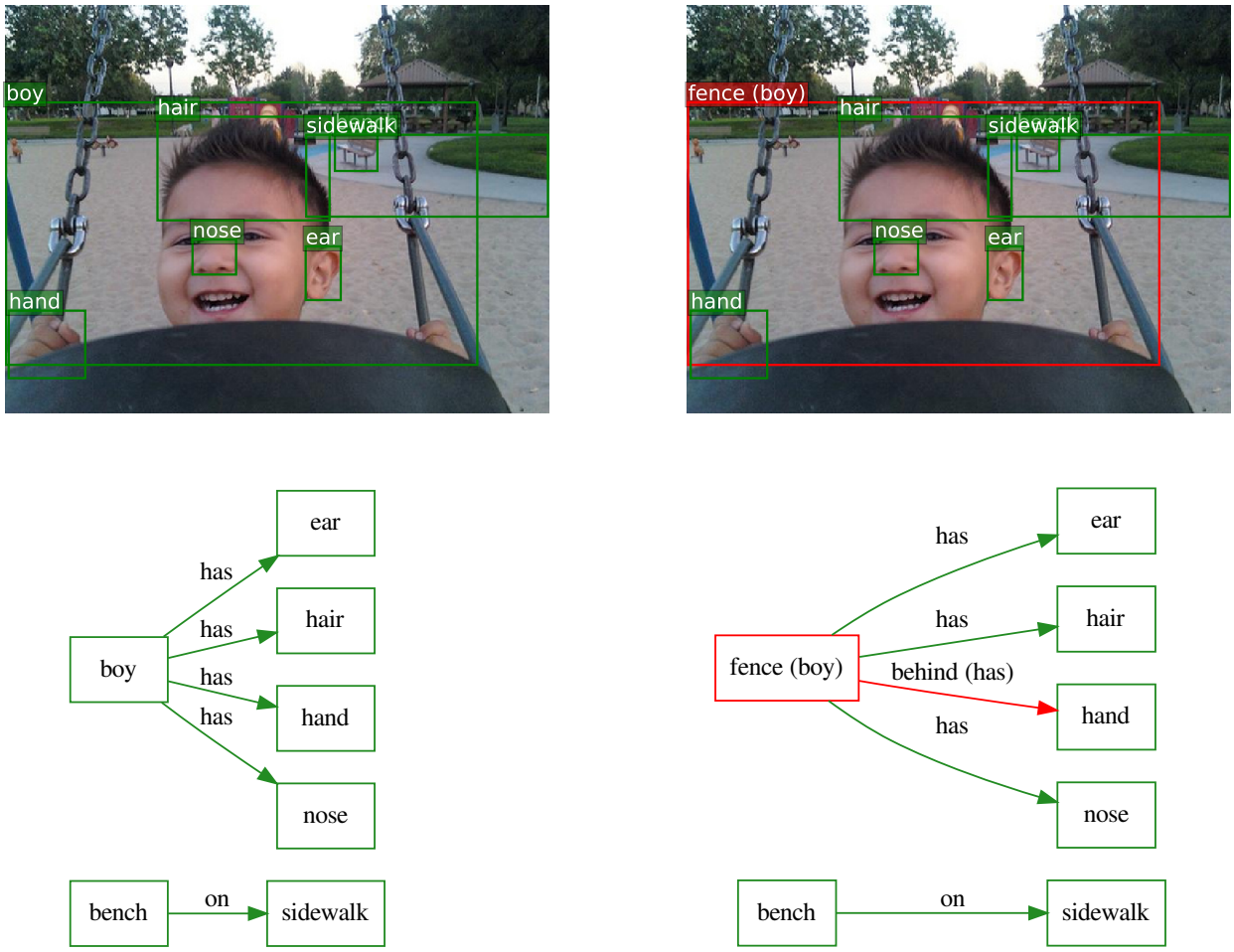


Figure 4.7: Example comparison of our method GB-NET (left) with KERN [78] (right). KERN misclassifies the boy as fence, which leads to the nonsensical triplets fence has ear, fence has nose, etc. Our method is less likely to make such meaningless predictions.

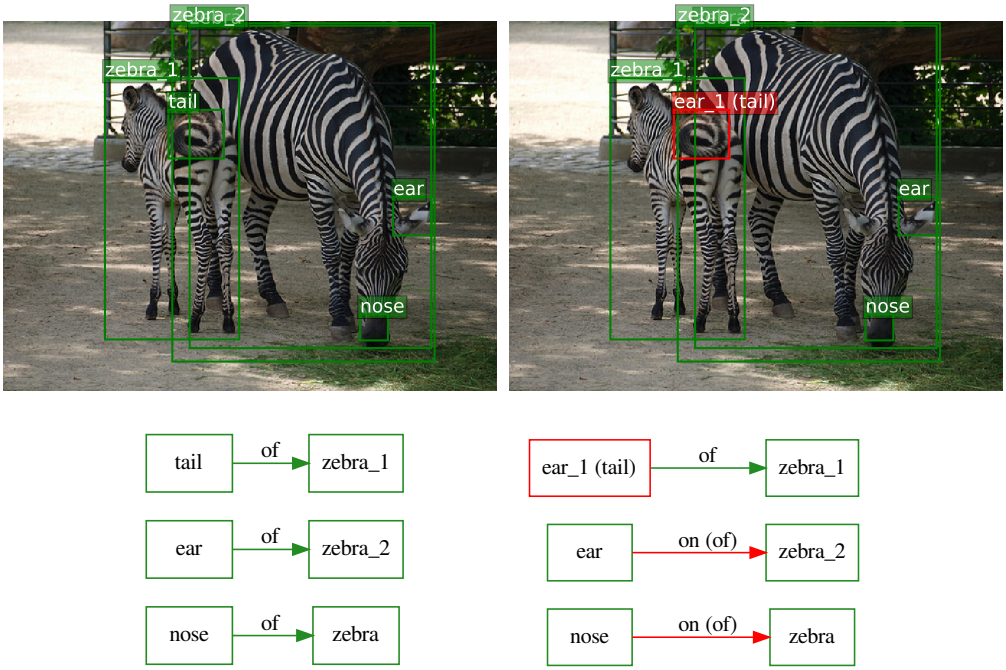


Figure 4.8: Example comparison of our method GB-NET (left) with KERN [78] (right). KERN predicts triplets such as ear on zebra and nose on zebra, etc., while our method predicts more semantically sound triplets ear of zebra and nose of zebra, reflecting the ownership relationship between the zebra and its body parts.

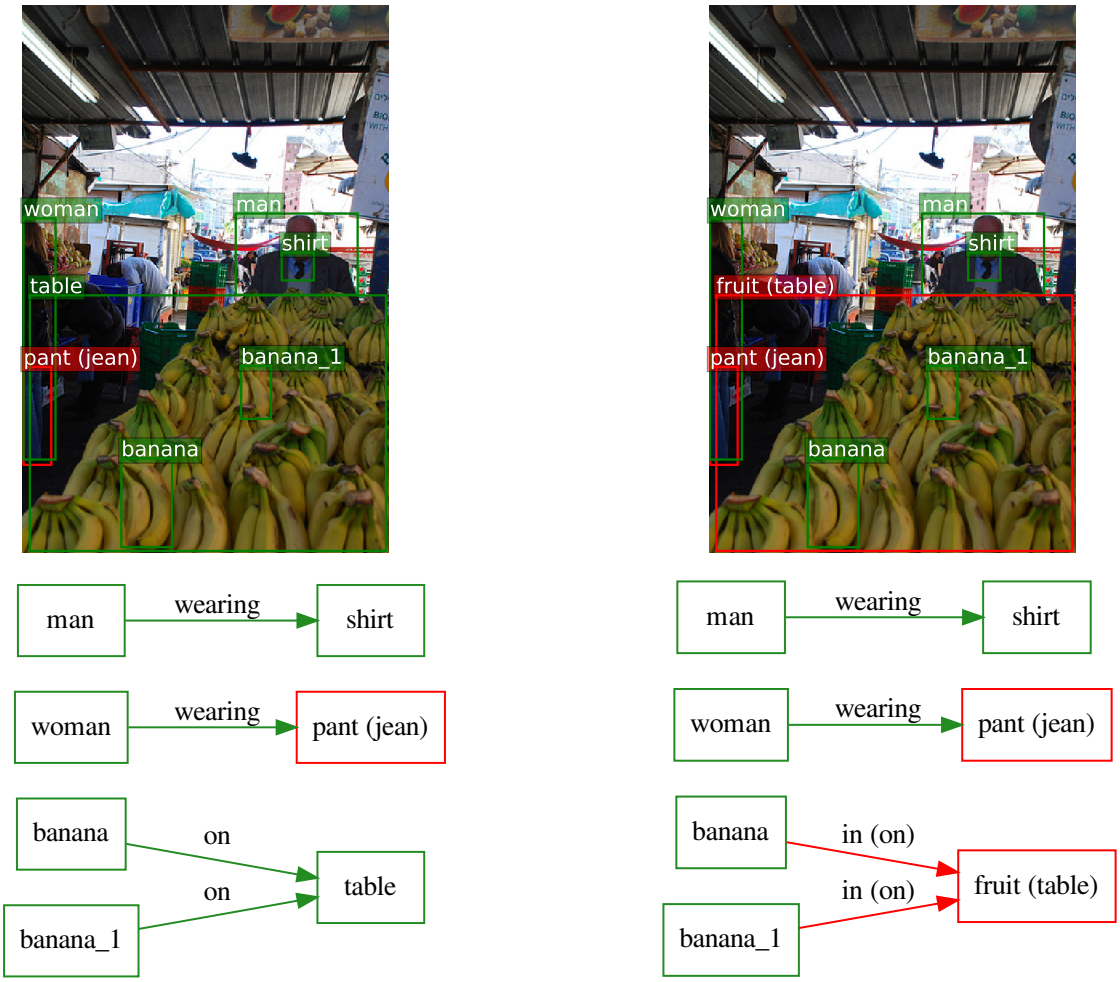


Figure 4.9: Example comparison of our method GB-NET (left) with KERN [78] (right). KERN misclassifies the table as fruit, possibly because it is entirely covered by fruites. But this leads to nonsensical triplet banana in fruit. Our method correctly classifies the table, which leads to a more commonsensical scene graph.

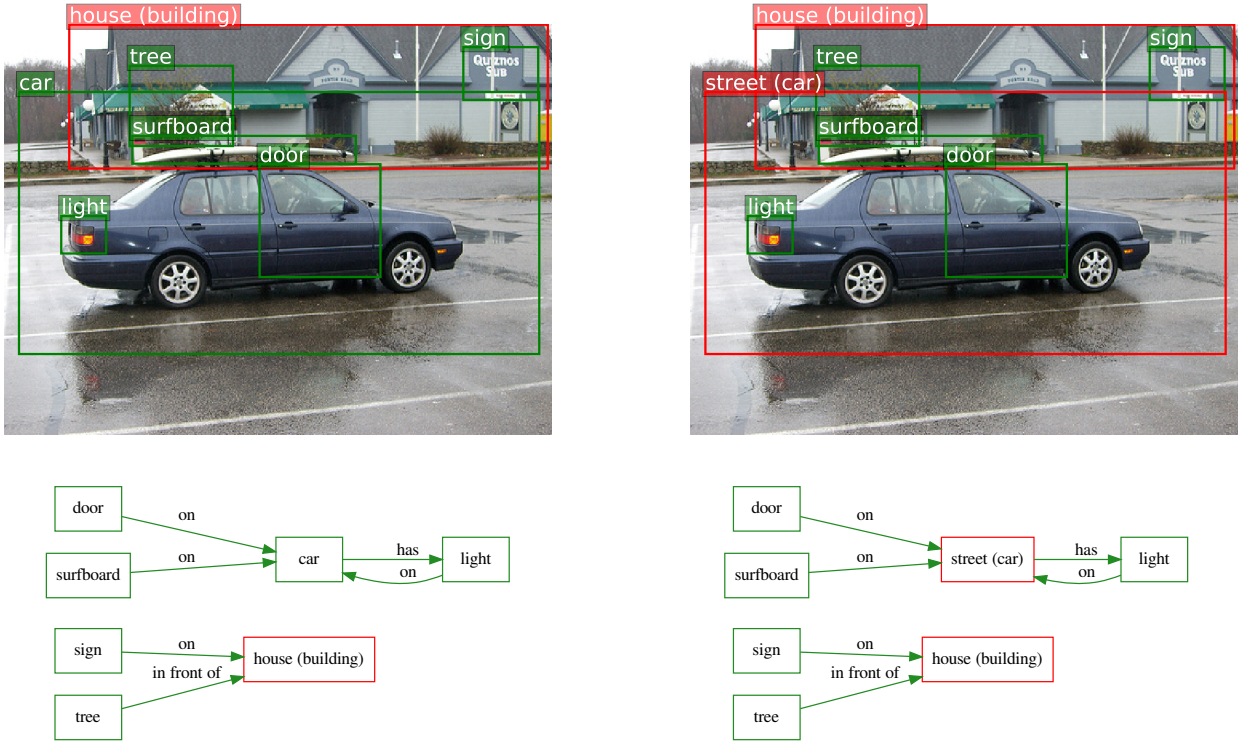


Figure 4.10: Example comparison of our method GB-NET (left) with KERN [78] (right). KERN misclassifies car as street, possibly because the bounding box is too loose and contains a large portion of the street. Our method is aware that door on street is not commonsensical, and hence predicts the more appropriate choice, *i.e.* car.

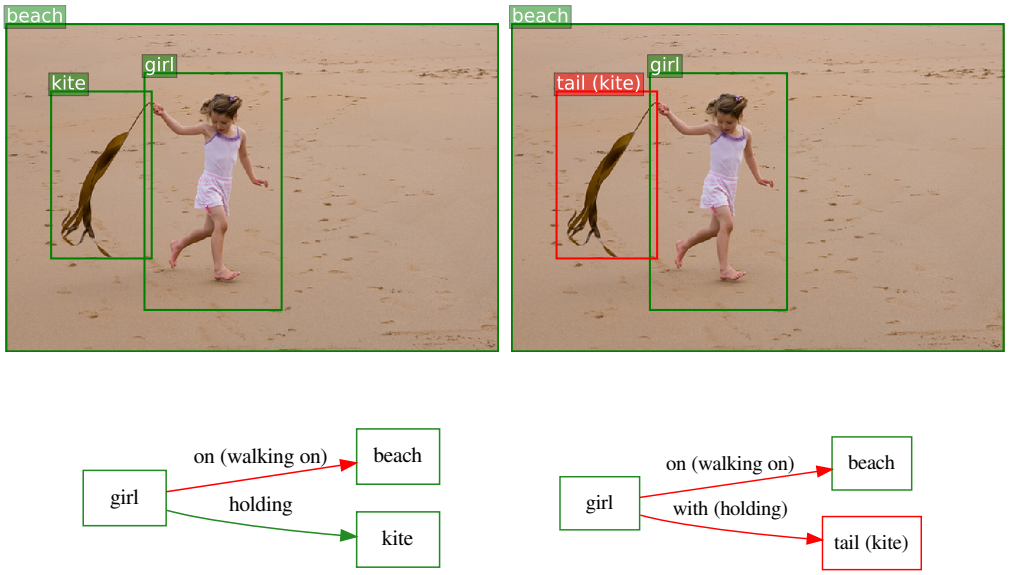


Figure 4.11: Example comparison of our method GB-NET (left) with KERN [78] (right). KERN misclassifies the kite as a tail, because it actually looks more like a tail. Our method predicts kite that is visually less clear, but leads to a more commonsensical graph overall.

4.8 Summary

We proposed a new method for Scene Graph Generation that incorporates external commonsense knowledge in a novel, graphical neural framework. We unified the formulation of scene graph and commonsense graph as two types of knowledge graph, which are fused into a single graph through a dynamic message passing and bridging algorithm. Our method iteratively propagates messages to update nodes, then compares nodes to update bridge edges, and repeats until the two graphs are carefully connected. Through extensive experiments, we showed our method outperforms the state of the art in various metrics.

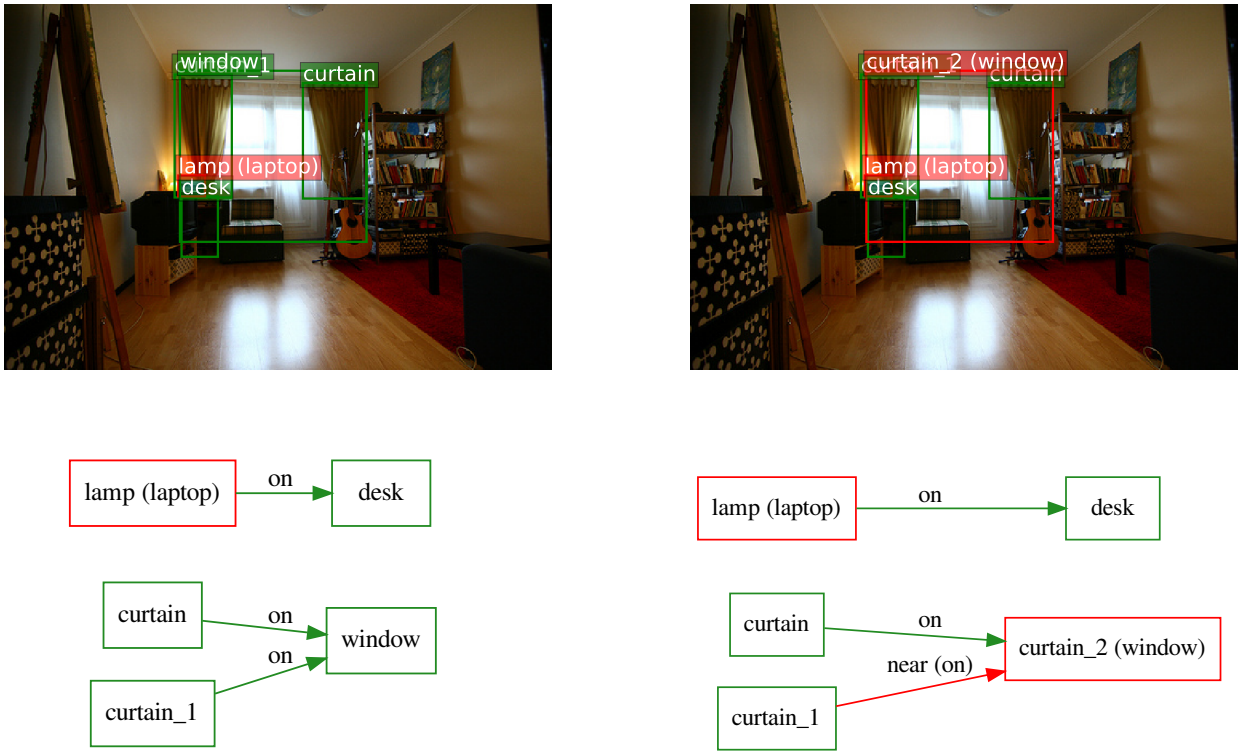


Figure 4.12: Example comparison of our method GB-NET (left) with KERN [78] (right). Our method correctly detects the two pieces of curtain on window, while KERN predicts the less appropriate triplet curtain on curtain, possibly because the bounding box of the window contains the curtain as well.

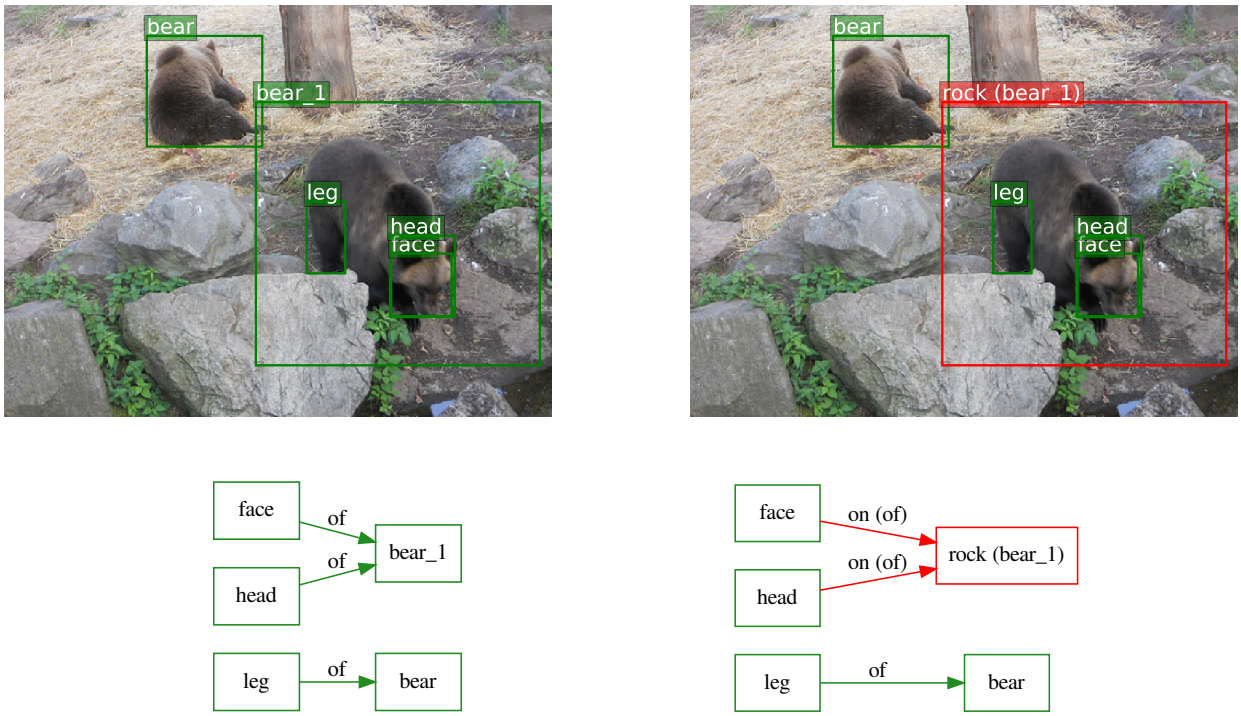


Figure 4.13: Example comparison of our method GB-NET (left) with KERN [78] (right). KERN misclassifies the bear as rock, possibly due to the too loose bounding box that includes rocks as well. This leads to nonsensical triplets such as face on rock and head on rock, while our method produces more likely and accurate triplets face of bear and head of bear.

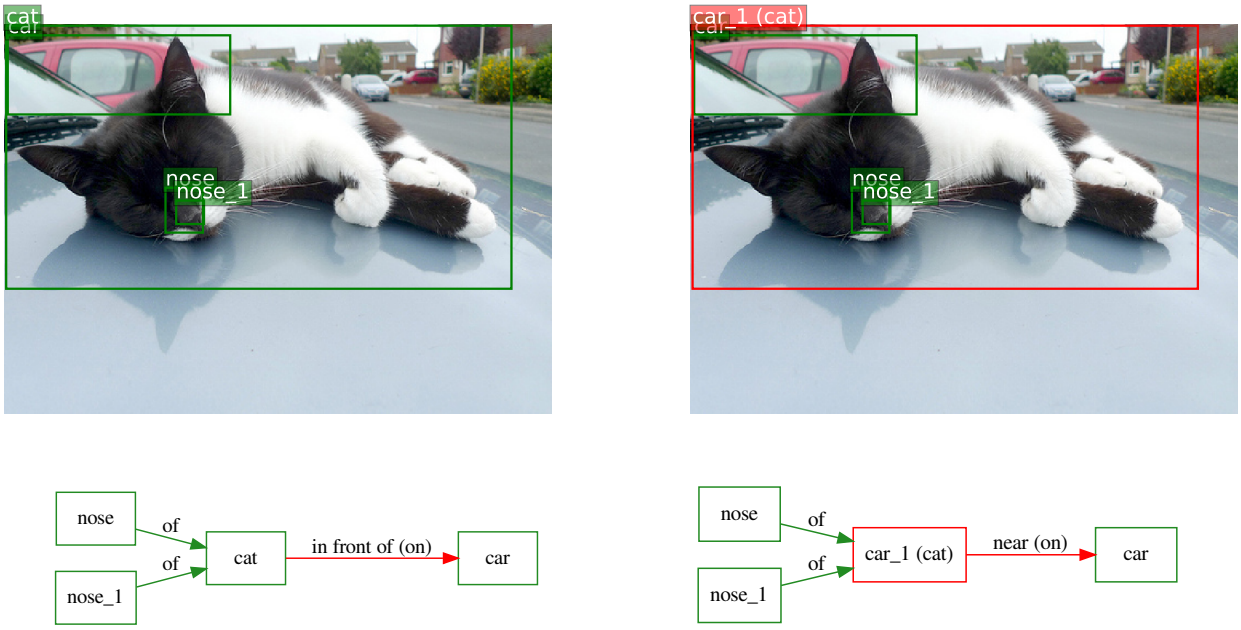


Figure 4.14: Example comparison of our method GB-NET (left) with KERN [78] (right). KERN misclassifies the cat as a car, possibly because the bounding box is too loose and covers a large area of both cars. Our method exploits the fact that cars are unlikely to have noses.

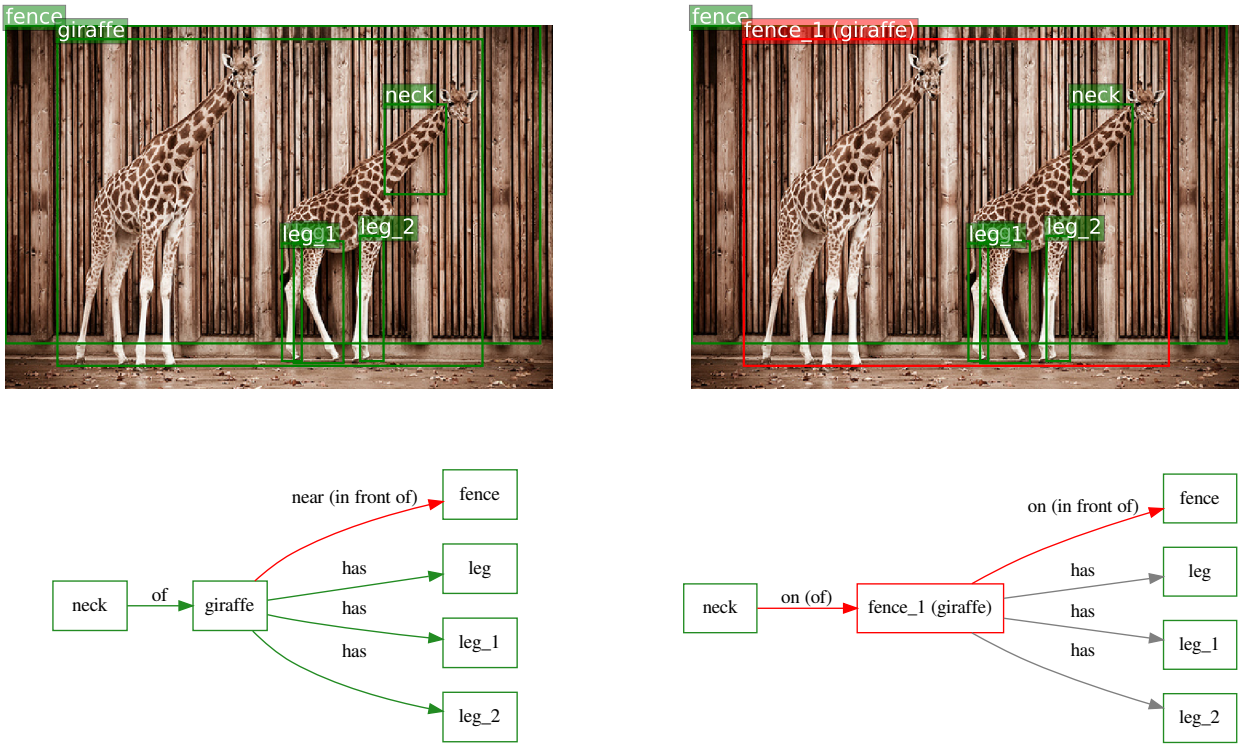


Figure 4.15: Example comparison of our method GB-NET (left) with KERN [78] (right). KERN misclassifies the giraffe as a fence, leading to nonsensical triplets such as fence on fence, fence has leg, etc. Our method avoids such inappropriates compositions.

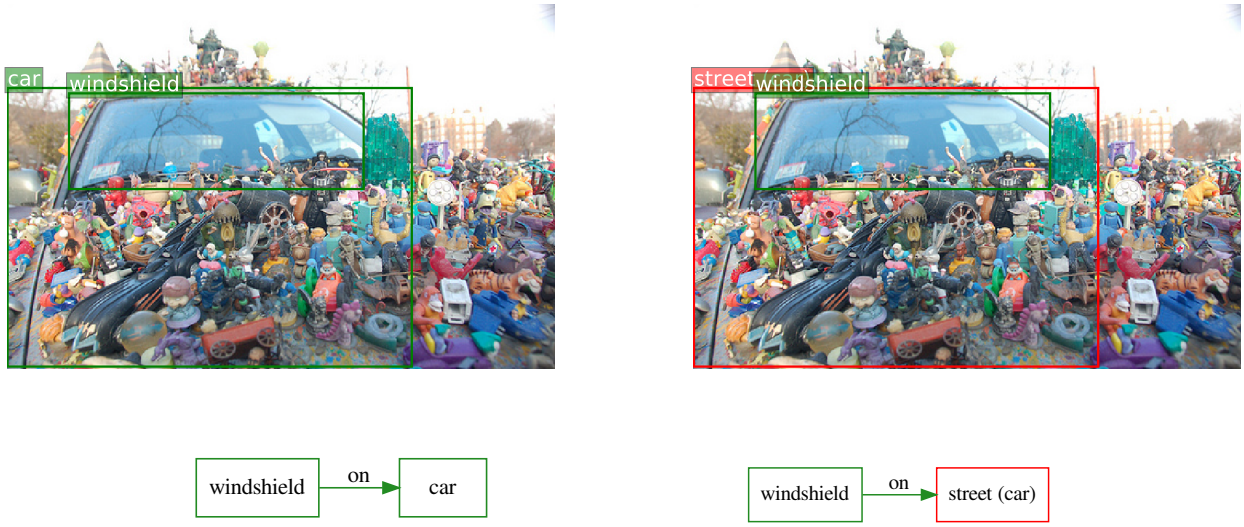


Figure 4.16: Example comparison of our method GB-NET (left) with KERN [78] (right). KERN misclassifies car as street due to the extreme occlusion, while our method exploits the fact that cars are more likely to have windshields than streets.

Table 4.1: Evaluation in terms of mean and overall triplet recall, at top 50 and top 100, with and without Graph Constraint (GC), for the three tasks of SGGEN, SGCLS and PREDCLS. Numbers are in percentage. All baseline numbers were borrowed from [78]. Top two methods for each metric is shown in **bold** and *italic* respectively.

Task	Metric	GC	Method					
			IMP+	FREQ	SMN	KERN	GB-NET	GB-NET-β
SGGEN	mR@50	Y	3.8	4.3	5.3	*6.4*	6.1	**7.1**
		N	5.4	5.9	9.3	*11.7*	9.8	**11.7**
	mR@100	Y	4.8	5.6	6.1	7.3	*7.3*	**8.5**
		N	8.0	8.9	12.9	*16.0*	14.0	**16.6**
	R@50	Y	20.7	23.5	**27.2**	*27.1*	26.4	26.3
		N	22.0	25.3	*30.5*	**30.9**	29.4	29.3
	R@100	Y	24.5	27.6	**30.3**	29.8	*30.0*	29.9
		N	27.4	30.9	*35.8*	**35.8**	35.1	35.0
SGCLS	mR@50	Y	5.8	6.8	7.1	9.4	*9.6*	**12.7**
		N	12.1	13.5	15.4	19.8	*21.4*	**25.6**
	mR@100	Y	6.0	7.8	7.6	10.0	*10.2*	**13.4**
		N	16.9	19.6	20.6	26.2	*29.1*	**32.1**
	R@50	Y	34.6	32.4	35.8	36.7	**38.0**	*37.3*
		N	43.4	40.5	44.5	45.9	**47.7**	*46.9*
	R@100	Y	35.4	34.0	36.5	37.4	**38.8**	*38.0*
		N	47.2	43.7	47.7	49.0	**51.1**	*50.3*
PREDCLS	mR@50	Y	9.8	13.3	13.3	17.7	*19.3*	**22.1**
		N	20.3	24.8	27.5	36.3	*41.1*	**44.5**
	mR@100	Y	10.5	15.8	14.4	19.2	*20.9*	**24.0**
		N	28.9	37.3	37.9	49.0	*55.4*	**58.7**
	R@50	Y	59.3	59.9	65.2	65.8	**66.6**	*66.6*
		N	75.2	71.3	81.1	81.9	**83.6**	*83.5*
	R@100	Y	61.3	64.1	67.1	67.6	**68.2**	*68.2*
		N	83.6	81.2	88.3	88.9	**90.5**	*90.3*

Table 4.2: Ablation study on Visual Genome. All numbers are in percentage, and graph constraint is enforced.

Method	SGGEN				PREDCLS			
	mR@50	mR@100	R@50	R@100	mR@50	mR@100	R@50	R@100
No Knowledge	5.5	6.6	25.3	28.8	15.4	16.8	62.5	64.5
$T = 1$	5.6	6.7	24.9	28.5	15.6	17.1	62.1	64.2
$T = 2$	5.7	6.9	26.1	29.7	18.2	19.7	66.7	68.4
GB-NET	**6.1**	**7.3**	**26.4**	**30.0**	**18.2**	**19.7**	**67.0**	**68.6**

Table 4.3: Time and memory cost of our method compared to the state of the art

Method	Test time (sec/image)	Train time (min/epoch)	# parameters (million)
KERN [78]	0.79	401.2	405.2
GB-NET	0.52	191.6	444.6

Chapter 5: Learning Visual Commonsense with Graph-Based Representations

In the previous chapter, we studied how to use an external knowledge graph to incorporate human-like commonsense within the process of graph-based scene understanding. Nevertheless, relying on an external source of knowledge is not ideal, as it requires manual work to produce such knowledge graphs, and existing sources are incomplete and noisy. In this chapter, we propose the first method to acquire visual commonsense such as affordance and intuitive physics automatically from data, and use that to improve the robustness of scene understanding. To this end, we extend Transformer models to incorporate the structure of scene graphs, and train our Global-Local Attention Transformer on a scene graph corpus. Once trained, our model can be applied on any scene graph generation model and correct its obvious mistakes, resulting in more semantically plausible scene graphs. Through extensive experiments, we show our model learns commonsense better than any alternative, and improves the accuracy of state-of-the-art scene graph generation methods. This chapter including all images, figures, tables, equations, and text is based on a recently published collaborative work [93].

5.1 Introduction

As we discussed in Chapter 4, there is a growing interest in incorporating commonsense reasoning and background knowledge into the process of visual recognition and scene understanding [40, 33, 94, 95, 9]. In Scene Graph Generation (SGG), for instance, external knowledge bases [79] and dataset statistics [78, 58] have been utilized to improve the accuracy of entity (object) and predicate (relation) recognition. The effect of these techniques is usually to correct obvious perception errors, and replace with more plausible alternatives. For instance, Figure 5.1 (top) shows an SGG

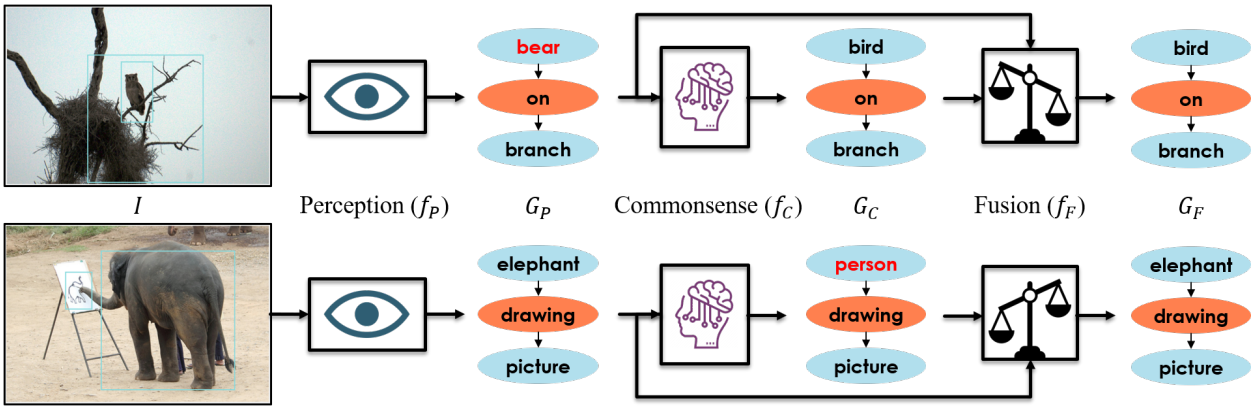


Figure 5.1: Overview of the proposed method: We propose a commonsense model that takes a scene graph generated by a perception model and refines that to make it more plausible. Then a fusion module compares the perception and commonsense outputs and generates a final graph, incorporating both signals.

model mistakenly classifies a `bird` as a `bear`, possibly due to the dim lighting and small object size. However, a model that mimics human commonsense can correctly predict `bird`, because `bear on branch` is a less common situation, less aligned with intuitive physics, and contrary to animal behavior.

Nevertheless, existing methods to incorporate commonsense into the process of visual recognition have two major limitations. Firstly, they rely on an external source of commonsense, such as crowd-sourced or automatically mined commonsense rules, which tend to be incomplete and inaccurate [79], or statistics directly gathered from training data, which are limited to simple heuristics such as co-occurrence frequency [78]. In this chapter, we propose the first method to learn graphical commonsense automatically from a scene graph corpus, which does not require external knowledge, and **acquires** commonsense by learning complex, structured patterns beyond simple heuristics.

Secondly, most existing methods are strongly vulnerable to data bias as they integrate data-driven commonsense knowledge into data-driven neural networks. For instance, the commonsense model in Figure 5.1 (bottom) mistakes the `elephant` for a `person`, in order to avoid the bizarre triplet `elephant drawing picture`, while the `elephant` is quite clear visually, and the perception model already recognizes it correctly. None of the existing efforts to equip scene under-

standing with commonsense have studied the fundamental question of whether to trust perception or commonsense, *i.e.*, what you see versus what you expect. In this chapter, we propose a way to disentangle perception and commonsense into two separately trained models, and introduce a method to exploit the disagreement between those two models to achieve the best of both worlds.

To this end, we first propose a mathematical formalization of visual commonsense, as a problem of auto-encoding perturbed scene graphs. Based on the new formalism, we propose a novel method to learn visual commonsense from annotated scene graphs. We extend recently successful transformers [96] by adding local attention heads to enable them to encode the structure of a scene graph, and we train them on a corpus of annotated scene graphs to predict missing elements of a scene via a masking framework similar to BERT [97]. As illustrated in Figure 5.2, our commonsense model learns to use its experience to imagine which entity or predicate could replace the mask, considering the structure and context of a given scene graph. Once trained, it can be stacked on top of any perception (*i.e.,* SGG) model to correct nonsensical mistakes in the generated scene graphs.

The output of the perception and commonsense models can be seen as two generated scene graphs with potential disagreements. We devise a fusion module that takes those two graphs, along with their classification confidence values, and predicts a final scene graph that reflects both perception and commonsense knowledge. The degree to which our fusion module trusts each input varies for each image, and is determined based on the estimated confidence of each model. This way, if the perception model is uncertain about the `bird` due to darkness, the fusion module relies on the commonsense more, and if perception is confident about the `elephant` due to its clarity, the fusion module *trusts its eyes*.

We conduct extensive experiments on the Visual Genome datasets [54], showing (1) The proposed GLAT model outperforms existing transformers and graph-based models in the task of commonsense acquisition; (2) Our model learns various types of commonsense that are absent in SGG models, such as object affordance and intuitive physics; (3) The proposed model is robust to dataset bias, and shows commonsensical behavior even in rare and zero-shot scenarios; (4) The proposed

GLAT and Fusion mechanisms can be applied on any SGG method to correct their mistakes and improve their accuracy. The main contributions of this chapter are the following:

- We propose the first method for learning structured visual commonsense, Global-Local Attention Transformer (GLAT), which does not require any external knowledge, and outperforms conventional transformers and graph-based networks.
- We propose a cascaded fusion architecture for Scene Graph Generation, which disentangles commonsense reasoning from visual perception, and integrates them in a way that is robust to the failure of each component.
- We report experiments that showcase our model's unique ability of learning commonsense without picking up dataset bias, and its utility in downstream scene understanding.

5.2 Related Work

5.2.1 Commonsense in computer vision

Incorporating commonsense knowledge has been explored in various computer vision tasks such as object recognition [62, 52, 53], object detection [94], semantic segmentation [98], action recognition [40], visual relation detection [33], scene graph generation [78, 58, 79], and visual question answering [99, 100]. There are two aspects to study about these methods: where their commonsense comes from, and how they use it.

Most methods either adopt an external curated knowledge base such as ConceptNet [101, 79, 52, 53, 98, 99], or acquire commonsense automatically by collecting statistics over an often annotated corpus [62, 78, 58, 94, 100, 33]. Nevertheless, the former group are limited to incomplete external knowledge, and the latter are based on ad-hoc, hard-coded heuristics such as the co-occurrence frequency of categories. Our method is the first to formulate visual commonsense as a machine learning task, and train a graph-based neural network to solve it. There are a third group of works that focus on a particular type of commonsense by designing a specialized model,

such as intuitive physics [102], or object affordance [103]. We put forth a more general framework that includes but is not limited to physics and affordance, by exploiting scene graphs as a versatile semantic representation. The most similar to our work is [104], which only models object co-occurrance patterns, while we also incorporate object relationships and scene graph structure.

When it comes to utilizing commonsense, existing methods integrate it within the inference pipeline, either by retrieving a set of relevant facts from a knowledge base and feeding as additional features to the model [99, 79, 100], or by employing a graph-based message propagation process to embed the structure of the knowledge graph within the intermediate representations of the model [62, 40, 78, 52, 53]. Some other methods distill the knowledge during training through auxiliary objectives, making the inference simple and free of external knowledge [98, 33]. Nevertheless, in all those approaches, commonsense is seamlessly infused into the model and cannot be disentangled. This makes it hard to study and evaluate commonsense and perception separately, or control their influence. Few methods have modeled commonsense as a standalone module which is late-fused into the prediction of the perception model [94, 58]. Yet, we are the first to devise separate perception and commonsense models, and adaptively weigh their importance based on their confidence, before fusing their predictions.

5.2.2 Commonsense in scene graph generation

Zellers *et al.* [58] were the first to explicitly incorporate commonsense into the process of scene graph generation. They biased predicate classification logits using a pre-computed frequency prior that is a static distribution, given each entity class pair. Although this significantly improved their overall accuracy, the improvement is mainly due to the fact that they favor frequent triplets over others, which is statistically rewarding. Even if their model classifies the relation between a `person` and a `hat` as `holding`, their frequency bias would most likely change that to `wearing`, which is more frequent.

More recently, Chen *et al.* [78] employed a less explicit way to incorporate the frequency prior within the process of entity and predicate classification. They embed the frequencies into the edge

weights of their inference graph, and utilize those weights within their message propagation process. This improves the results especially on less frequent predicates, since it less strictly enforces the statistics on the final decision. However, this way commonsense is integrated implicitly into the SGG model and cannot be probed or studied in isolation. We remove the adverse effect of statistical bias while keeping the commonsense model disentangled from perception.

Gu *et al.* [79] exploits ConceptNet [101] rather than dataset statistics, which is a large-scale knowledge graph comprising relational facts about concepts, *e.g.* `dog is-a animal` or `fork is-used-for eating`. Given each detected object, they retrieve ConceptNet facts involving that object class, and employ a recurrent neural net and an attention mechanism to encode those facts into the object features, before classifying objects and predicates. Nevertheless, ConceptNet is not exhaustive, since it is extremely hard to compile all commonsense facts. Our method does not depend on a limited source of external knowledge, and acquires commonsense automatically, via a generalizable neural network.

5.2.3 Transformers and graph-based neural networks

Transformers were originally proposed to replace recurrent neural networks for machine translation, by stacking several layers of multi-head attention [96]. Ever since, transformers have been successful in various vision and language tasks [97, 105, 1]. Particularly, BERT [97] randomly replaces some words from a given sentence with a special `MASK` token and tries to reconstruct those words. Through this self-supervised game, BERT acquires natural language, and can transfer its language knowledge to perform well in other NLP tasks. We use a similar self-supervised strategy to learn to complete missing pieces of a scene graph. Rather than language, our model acquires the ability to imagine a scene in a structured, semantic way, which is a hallmark of human commonsense.

Transformers treat their input as a set of tokens, and discard any form of structure among them. To preserve the order of tokens in a sentence, BERT augments the initial embedding of each token with a position embedding before feeding into transformers. Scene graphs, on the other hand, have

a more complex structure that cannot be embedded in such a trivial way. Recently, Graph-based Neural Networks (GNN) have been successful to encode graph structures into node representations, by applying several layers of neighborhood aggregation. More specifically, each layer of a GNN represents each node by a trainable function that takes the node as well as its neighbors as input. Graph convolutional nets [47], gated graph neural nets [89], and graph attention nets [106] all implement this idea with different computational models for neighborhood aggregation. GNNs have been widely utilized for scene graph generation by incorporating context [37, 50, 57], but we are the first to exploit GNNs to learn visual commonsense.

We adopt graph attention nets due to their similarity to transformers in using attention. The main difference of graph attention nets to transformers is that instead of representing each node by an attention over all other nodes, they only compute an attention over immediate neighbors. Inspired by that, we use a BERT-like transformer network, but replace half of its attention heads by local attention, simply by enforcing the attention between non-neighbor nodes to zero. Through ablation experiments in Section 5.4, we show the proposed Global-Local Attention Transformers (GLAT) outperforms conventional transformers, as well as widely used graph-based models such as graph convolution nets and graph attention nets.

5.3 Method

In this section, we first formalize the task, and propose a novel formulation of visual commonsense in connection with visual perception. We then provide an overview of the proposed architecture (Figure 5.1), followed by an in-depth description of each proposed module.

We define a scene graph as $G = (\mathcal{N}_e, \mathcal{N}_p, \mathcal{E}_s, \mathcal{E}_o)$, where $\mathcal{N}_e$ is a set of entity nodes, $\mathcal{N}_p$ is a set of predicate nodes, $\mathcal{E}_s$ is a set of edges from each predicate to its subject (which is an entity node), and $\mathcal{E}_o$ is a set of edges from each predicate to its object (that also is an entity node). Each entity node is represented with an entity class $c_e \in C_e$ and a bounding box $b \in [0, 1]^4$, while each predicate node is represented with a predicate class $c_p \in C_p$ and is connected to exactly one subject and one object. Note that this formulation of scene graph is slightly different from the

conventional one [37], as we formulate predicates as nodes rather than edges. This tweak does not cause any limitation since every scene graph can be converted from the conventional representation to our representation. However, this formulation allows multiple predicates between the same pair of entities, and it also enables us to define a unified attention over all nodes no matter entity or predicate.

Given a training dataset with many images $I \in [0,1]^{h \times w \times c}$ paired with ground truth scene graphs G_T, our goal is to train a model that takes a new image and predicts a scene graph that maximizes $p(G|I)$. This is equivalent of maximizing $p(I|G)p(G)$, which breaks the problem into what we call *perception* and *commonsense*. In our proposed intuition, commonsense is the mankind's ability to predict which situations are possible and which are not, or in other words, what makes *sense* and what does not. This can be seen as a prior distribution $p(G)$ over all possible situations in the world, represented as scene graphs. Perception, on the other hand, is the ability to form symbolic belief from raw sensory data, which are respectively G and I in our case. Although the goal of computer vision is to solve the Maximum a Posteriori (MAP) problem (maximizing $p(G|I)$), neural nets often fail to estimate the posterior, unless the prior is explicitly enforced in the model definition [107]. This is while in computer vision, the prior is often overlooked, or inaccurately considered to be a uniform distribution, making MAP equivalent to Maximum Likelihood (ML), *i.e.*, finding G that maximizes $p(I|G)$ [108].

We propose the first method to explicitly approximate the MAP inference by devising an explicit prior model (commonsense). Since posterior inference is intractable, we propose a two-stage framework as an approximation: We first adopt any off-the-shelve SGG model as the *perception model*, which takes an input image and produces a perception-driven scene graph, G_P, that approximately maximizes the likelihood. Then we propose a *commonsense model*, which takes G_P as input, and produces a commonsense-driven scene graph, G_C, to approximately maximize the

posterior, i.e.,

$$G_P = f_P(I) \approx \arg\max_G p(I|G), \tag{5.1}$$

$$G_C = f_C(G_P) \approx \arg\max_G p(I|G)p(G), \tag{5.2}$$

where f_P and f_C are the perception and commonsense models. The commonsense model can be seen as a graph-based extension of denoising autoencoders [109], which evidently can learn the generative distribution of data [110, 111], that is $p(G)$ in our case. Accordingly, f_C can take any scene graph as input and produce a more plausible graph by only slightly changing the input. A key design choice here is the fact that f_C does not take the image as input. Otherwise, it would be hard to ensure it is purely learning commonsense and not perception.

Ideally, G_C is the best decision to make, since it maximizes the posterior distribution. However, in practice autoencoders tend to under-represent long-tailed distributions and only capture the modes. This means the commonsense model may fail to predict less common structures, in favor of more statistically rewarding alternatives. To alleviate this problem, we propose a *fusion module* that takes G_P and G_C as input, and outputs a fused scene graph, G_F, which is the final output of our system. This can be seen as a decision-making agent that has to decide how much to trust each model, based on how confident they are.

Figure 5.1 illustrates an overview of the proposed architecture. In the rest of this section, we elaborate each module in detail.

5.3.1 Global-Local Attention Transformers

We propose the first graph-based visual commonsense model, which learns a generative distribution over the semantic structure of real-world scenes, through a denoising autoencoder framework. Inspired by BERT [97], which reconstructs masked tokens in a sentence through stacked layers of multi-head attention, we propose Global-Local Attention Transformers (GLAT) that take a graph with masked nodes as input, and reconstructs the missing nodes. Figure 5.2 illustrates how

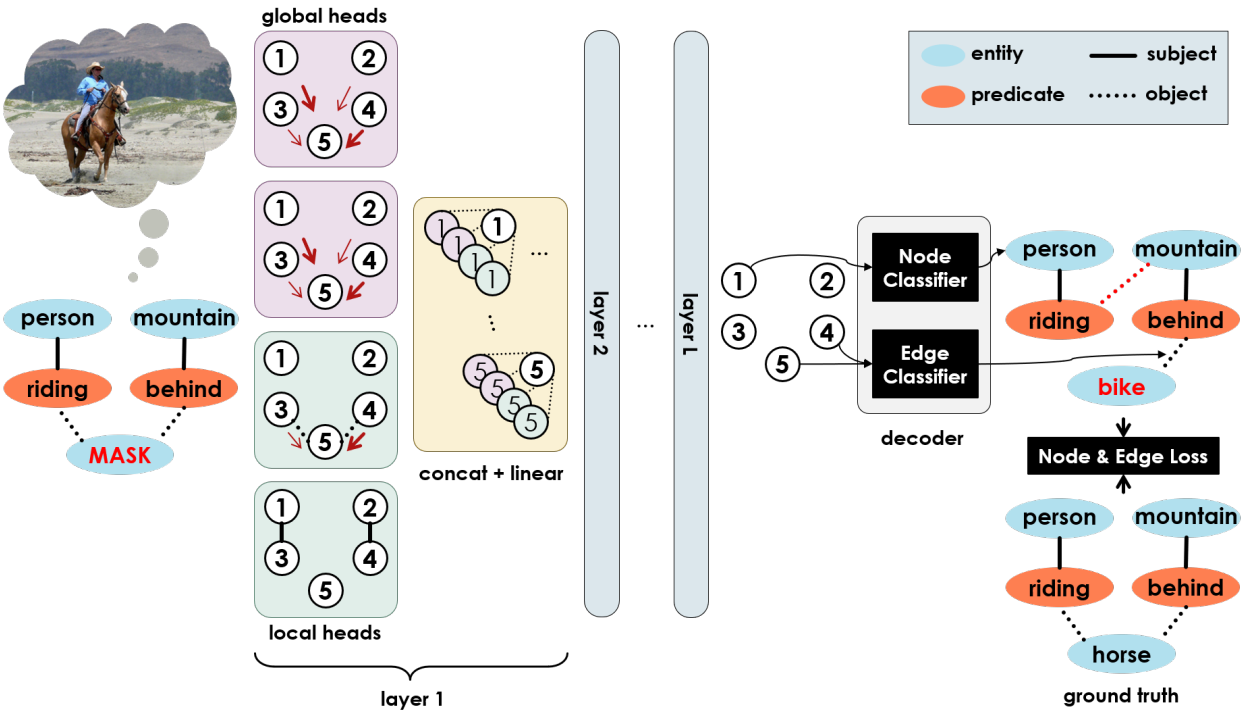


Figure 5.2: The proposed Global-Local Attention Transformer (GLAT), and its training framework: We augment transformers with local attention heads to help them encode the structure of scene graphs within node embeddings. The decoder takes the embeddings of a perturbed scene graph and reconstructs the correct scene graph without having access to the image. Note this figure only shows the commonsense block of our overall pipeline shown in Figure 5.1.

GLAT works. Given an input scene graph G_P, we represent node i as a one-hot vector $x_i^{(0)}$, that includes entity and predicate categories, as well as a special `MASK` class. We stack node representations as rows of a matrix $X^{(0)}$ for notation purposes.

GLAT takes $X^{(0)}$ as input and represents each node by encoding the structure and context. To this end, it applies L layers of multi-head attention on the input nodes. Each layer l creates new node representations $X^{(l)}$, by applying a linear layer on the concatenated output of that layer's attention heads. More specifically,

$$X^{(l)} = \underset{h \in \mathcal{H}_l}{\text{concatenate}} \left[h(X^{(l-1)}) \right] \times W_l + b_l, \tag{5.3}$$

where $\mathcal{H}_l$ is the set of attention heads for layer l, W_l and b_l are trainable fusion weights and bias for that layer, and the concatenation operates along columns. We use two types of attention head,

namely global and local. Each node can attend to all other nodes through global attention, while only its neighbors through local attention. We further divide local heads based on the type of edge they use, in order to differentiate the way subjects and objects interact with predicates, and vice versa. Therefore, we can write:

$$\mathcal{H}_l = \mathcal{H}_l^G \cup \mathcal{H}_l^{LS} \cup \mathcal{H}_l^{LO}. \tag{5.4}$$

All heads within each subset are identical, except they have distinct parameters that are initialized and trained independently. Each global head h^G operates as a typical self-attention would:

$$h^G(X) = \left[q(X)^T k(X)\right] v(X), \tag{5.5}$$

where q, k, v are query, key, and value heads, each a fully connected network, typically (but not necessarily) with a single linear layer. A local attention is the same, except queries can only interact with keys of their immediate neighbor nodes. For instance in subject heads,

$$h^{LS}(X) = \left[q(X)^T k(X) \odot A_s\right] v(X), \tag{5.6}$$

where A_s is the adjacency matrix of subject edges, which is 1 between from each predicate to its subject and vice versa, and 0 elsewhere. We similarly define A_o and h^{LO} for object edges.

Once we get contextualized, structure-aware representations $x_i^{(L)}$ for each node i, we devise a simple decoder to generate the output scene graph G_C, using a fully connected network that classifies each node to an entity or predicate class, and another fully connected network that classifies each pair of nodes into an edge type (subject, object or no edge). We train the encoder and decoder end-to-end, by randomly adding noise to annotated scene graphs from Visual Genome, feeding the noisy graph to GLAT, reconstructing nodes and edges, and comparing each with the original scene graph before perturbation. We train the network using two cross-entropy loss terms on the node and edge classifiers. The details of training including the perturbation process are explained

in Section 5.4.1.

5.3.2 Fusing Perception and Commonsense

The perception and commonsense models each predict the output node categories using a classifier that computes a probability distribution over all classes by applying a softmax on its logits. The class with highest probability is chosen and assigned a confidence score equal to its softmax probability. More specifically, node i from G_P has a logit vector L_i^P that has $|C_e|$ or $|C_p|$ dimensions depending of whether it is an entity node or predicate node. Similarly node i from G_C has a logit vector L_i^C. Note that these two nodes correspond to the same entity or predicate in the image, since GLAT does not change the order of nodes. Then the confidence of each node can be written as

$$q_i^P = \max_j \frac{\exp(L_i^P[j])}{\sum_k \exp(L_i^P[k])}, \tag{5.7}$$

and similarly q_i^C is defined given L_i^C.

The fusion module takes each node of G_P and the corresponding node of G_C, and computes a new logit vector for that node, as a weighted average of L_i^P and L_i^C. The weights determine the contribution of each model in the final prediction, and thus have to be proportional to the confidence of each model. Therefore, we compute the fused logits as:

$$L_i^F = \frac{q_i^P L_i^P + q_i^C L_i^C}{q_i^P + q_i^C}. \tag{5.8}$$

Finally, a softmax is applied on L_i^F to compute the final classification distribution for node i.

5.4 Experiments

In this section, we describe our experiments on the Visual Genome (VG) dataset in detail. We first evaluate how well our GLAT model learns visual commonsense, by comparing it to other models on the task of masked scene graph reconstruction. Then we provide a statistical analysis of our model prediction to show the kinds of commonsense knowledge it acquires, and distinguish it from

bias. Next, we evaluate how effective GLAT and our fusion mechanism are for the downstream task of SGG, when applied on various perception models. We also provide several examples of how the commonsense model corrects the perceived output, and how the fusion model combines the two.

5.4.1 Implementation details

We train the perception and commonsense models separately using the ground truth scene graphs G_T from VG [54], particularly the version most widely used for SGG [37], which has 150 entity and 50 predicate classes. We then stack commonsense on top of perception and fine-tune it on VG, this time with actual scene graphs generated by perception, to adapt to the downstream task. The fusion module does not have trainable parameters and is thus only used during inference. We use the 75k VG scene graphs for training all models, and use the other 25k for test. We hold a small portion of the train set for validation. Our GLAT model (and other baselines when applicable) have 6 layers, each with 8 attention heads, and has a 300-dimensional representation for each node. While training GLAT, we randomly mask 30% of the nodes, which is the average number of nodes mistaken by a typical SGG model. We average the classification loss over all nodes and edges classified by the decoder, no matter masked or not. For fine-tuning and inference, we prune the output of the perception model before feeding to GLAT, by keeping the top 100 most confident predicates and all entities connected to those.

5.4.2 Evaluating commonsense

Once GLAT is trained, we evaluate it on the same task of reconstructing ground truth VG graphs that are perturbed by randomly masking 30% of their nodes. We evaluate the accuracy of our model in classifying the masked nodes, and report the accuracy (Table 5.1) separately for entity nodes and predicate nodes, as well as overall. This is a good measure of how well the model has learned commonsense, because it mimics mankind's ability to imagine what would a real-world scene look like, given some context. In Figure 5.2, for instance, given the fact that there is a person

riding something that is masked, we can immediately tell it is probably a bike, a motorcycle, or a horse. If we also know there is a mountain behind the masked object, and the masked object has a face and legs (not shown in the figure for brevity), then we can more certainly imagine it is a horse. By incorporating the global context of the scene, as well as the local structure of the graph, GLAT is able to effectively imagine the scene and predict the class of the entity or predicate that was masked, at a significantly higher accuracy compared to all baselines.

More specifically, we compare GLAT to: (1) A transformer [97] that is the same as our model, except it only has global heads; (2) A Graph Attention Net [106] which is also the same as our model, but only with local heads; and (3) A Graph Convolutional Network [47], which has only one local head at each layer, and the attention is fixed to be equal for all neighbors of each node. We also compare our method with the frequency prior used by Zellers *et al.* [58], which can only be applied for masked predicates, and simply predicts the most frequent predicate given its subject and object. As Table 5.1 shows, our method significantly outperforms all aforementioned baselines, which are a good representative of any existing method to learn semantic graph reconstruction.

To provide a better sense of the commonsense knowledge our model learns, we apply GLAT on the entire VG test set, using the procedure detailed below (Section 5.4.3), and collect its prediction statistics in a diverse set of situations. We elaborate using an example, shown in the top left cell of Table 5.2. Out of all triplets from all scene graphs produced by our model, we collect those triplets that match the certain template of `person [X] horse`, and show our sorted top 5 predictions in terms of frequency. The 5 predicates most often predicted by our method between a `person` and a `horse` are `on`, `riding`, `near`, `watching`, and `behind`. These are all possible interactions between a person and a horse, and all follow the affordance properties of both `person` and `horse`. Nevertheless, when we get the same statistics from the output of a state-of-the-art scene graph generation model (IMP [37]), we observe that it frequently predicts `person wearing horse`, which does not follow the affordance of `horse`. This can be attributed to the high frequency of `wearing` in VG annotation, which biases the IMP model, while our commonsense model is prone to such bias, and has learned affordances through the self-supervised training

Table 5.1: Ablation study on Visual Genome. All numbers are in percentage, and graph constraint is enforced

Method	Entity	Predicate	Both
Triplet Frequency [58]	-	44.4	-
Graph Convolutional Nets [47] (local-only, fixed attention)	8.7	43.4	19.7
Graph Attention Nets [106] (local-only)	12.0	45.0	22.3
Transformers [97] (global-only)	14.0	42.3	22.9
Global-Local Attention Transformers (ours)	**22.3**	**60.7**	**34.4**

framework.

Table 5.2 provides several more scenarios like this, demonstrating our proficiency in three types of commonsense: object affordance, intuitive physics, and object composition. As an example of physics, we choose the triplet template `[X] under bed`, and show that our model predicts plausible objects such as `pot`, `shoe`, `drawer`, `book`, and `sneaker`. This is while IMP predicts `bed under bed`, `counter under bed`, and `sink under bed`, which are all physically counter-intuitive. More interestingly, one of our frequent predictions, `book under bed`, is a composition that does not exist in training data, suggesting the knowledge acquired by GLAT is not merely a biased memory of frequent compositions in training data.

The last type of commonsense in our illustration is object composition, *i.e.*, the fact that certain objects are physical parts of other objects. For `[X] has ear`, we predict `head`, `cat`, `elephant`, `zebra`, and `person`, out of which `head has ear` and `person has ear` are not within the 10 most frequent triplets in training data that match the template. Yet our model frequently predicts them, demonstrating its unbiased knowledge. Not to mention, 4 out of 5 top predictions made by IMP are nonsensical.

5.4.3 Evaluating scene graph generation

Now that we showed the efficacy of GLAT in learning visual commonsense and correcting perturbed scene graphs, we apply and evaluate it on the downstream task of scene graph generation. We adopt existing SGG models as our perception model, and compare their output G_P, to the ones

Table 5.2: Prediction statistics of our method compared to IMP [37] in various situations, showcasing our model's commonsense knowledge, and its robustness to dataset bias. Each row is designated for a certain type of commonsense, and has three examples in three pairs of columns. Each pair of columns show the top 5 most frequent triplets matching a certain template from our model's prediction, compared to IMP. **Black** triplets are commonsensically correct, **red** triplets are wrong, **blue** are commonsensically correct but statistically rare in training data, and **green** are correct but never seen in training data.

	Template 1		Template 2		Template 3	
	IMP + GLAT	IMP	IMP + GLAT	IMP	IMP + GLAT	IMP
Object Affordance	person on horse person riding horse person near horse person watching horse person behind horse	person on horse person riding horse person near horse **person wearing horse** person behind horse	person has flower **person with flower** person near flower **person watching flower** **person holding flower**	**person on flower** person has flower person near flower **person wearing flower** **person holding flower**	person sitting on chair person sitting on bench **person sitting on seat** person sitting on stand **person sitting on rock**	person sitting on chair person sitting on bench **person sitting on table** **person sitting on seat** **person sitting on person**
Intuitive Physics	orange in basket fruit in basket banana in basket food in basket paper in basket	**basket in basket** **man in basket** **woman in basket** orange in basket fruit in basket	airplane above plane **airplane above car** **airplane above bird** **airplane above beach** **airplane above vehicle**	**airplane above table**	pot under bed shoe under bed drawer under bed **book under bed** sneaker under bed	**bed under bed** drawer under bed **counter under bed** **sink under bed** shoe under bed
Object Composition	**head has ear** cat has ear elephant has ear zebra has ear **person has ear**	**leg has ear** **head has ear** **ear has ear** **tree has ear** **nose has ear**	skier has head **skier has hand** skier has leg skier has arm **skier has face**	skier has leg skier has arm skier has pole **skier has person** **skier has tree**	leg of person leg of man leg of giraffe leg of zebra leg of elephant	**leg of leg** leg of man leg of person **leg of tree** **leg of head**

corrected by our commonsense model G_C, as well as the final output of our system after fusion G_F. We compare those 3 outputs for 3 different choices of perception model, all of which have competitive state-of-the-art performance. More specifically, we use Iterative Message Passing (IMP [37]) as a strong baseline that is not augmented by commonsense. We also use Stacked Neural Motifs (SNM [58]) that late-fuse a frequency prior with their output, and Knowledge-Embedded Routing Networks (KERN [78]) that encode frequency prior within their internal message passing.

To evaluate, we conventionally compute the mean recall of the top 50 (mR@50) and top 100 (mR@100) triplets predicted by each model. Each triplet is considered correct if the subject, predicate, and object are all classified correctly, and the bounding box of the subject and object have more than 50% overlap (intersection over union) with the ground truth. We compute the recall for the triplets of each predicate class separately, and average over classes. The aforementioned metrics are measured in 2 sub-tasks: (1) SGCLS is the main scenario where we classify entities and predicates given annotated bounding boxes. This way the performance is not limited by proposal quality. (2) PREDCLS provides the model with ground truth object labels, which helps evaluation

Table 5.3: The mean recall of our method compared to the state of the art on the task of scene graph generation, evaluated on the Visual Genome dataset [37], following the experiment settings of [58]. All baseline numbers were borrowed from [78], and all numbers are in percentage

Method	PREDCLS		SGCLS	
	mR@50	mR@100	mR@50	mR@100
IMP [37]	9.8	10.5	5.8	6.0
IMP + GLAT	11.1	11.9	6.2	6.5
IMP + GLAT + Fusion	**12.1**	**12.9**	**6.6**	**7.0**
SNM [58]	13.3	14.4	7.1	7.5
SNM + GLAT	13.6	14.6	7.3	7.8
SNM + GLAT + Fusion	**14.1**	**15.3**	**7.5**	**7.9**
KERN [78]	17.7	19.2	9.4	10.0
KERN + GLAT	17.6	19.1	9.3	10.0
KERN + GLAT + Fusion	**17.8**	**19.3**	**9.9**	**10.4**

focus on predicate recognition accuracy. Table 5.3 shows the full comparison of all methods on all metrics. We observe that GLAT improves the performance of IMP which does not have commonsense, but does not significantly change the performance of SNM and KERN which already use dataset statistics. However, our full model which uses both the output of the perception model as well as commonsense model consistently improves SGG performance.

Finally, we provide several examples in Figure 5.3 to illustrate how our commonsense model fixes perception errors in difficult scenarios, and improves the robustness of our model. To save space, we merge the three scene graphs predicted by the perception, commonsense, and fusion models into a single graph, and emphasize any node or edge where these three models disagree. In example (a), the `chair` is not fully visible, and the visible part does not visually show the action of `sitting`, thus the perception model incorrectly predicts `wearing`, which is likely to be also affected by the bias due to the prevalence of `wearing` annotations in Visual Genome. However, it is trivial for the commonsense model that the affordance of `chair` is `sitting`. The fusion module correctly prefers the output of the commonsense model, due to its higher confidence. In (b), the perception model mistakes the `head` of the `bird` for a `hat`, due to the complexity of the lighting and the similarity of foreground and background colors. This might be also affected by

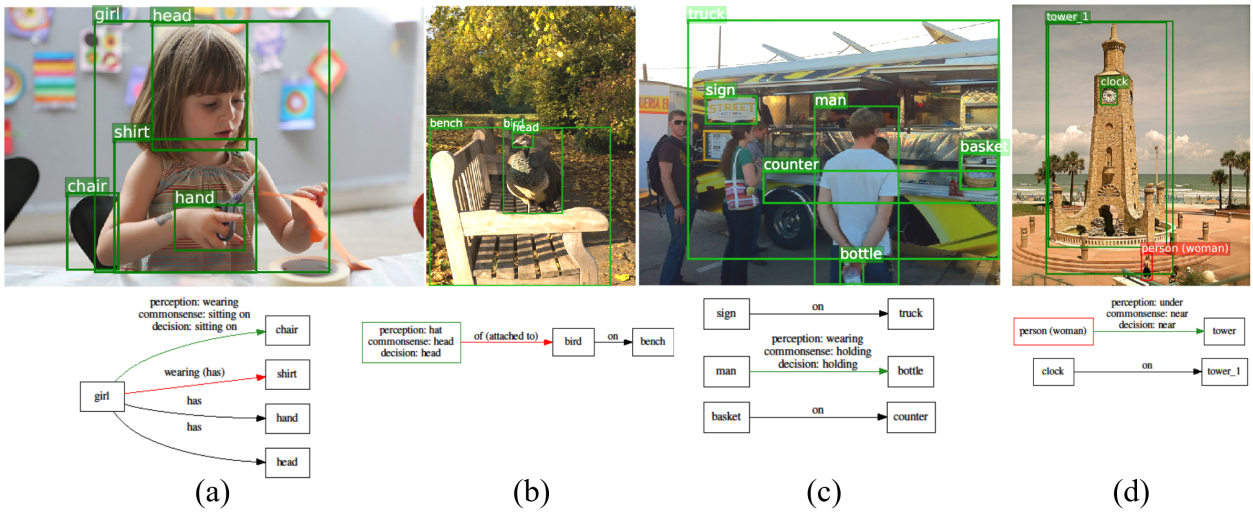


Figure 5.3: Example scene graphs generated by the perception, commonsense, and fusion modules, merged into one graph. Entities are shown as rectangular nodes and predicates are shown as directed edges from subject to object. For entities and predicates that are identically classified by the perception and commonsense model, we simply show the predicted label. But in cases where the perception and commonsense models disagree, we show both of their predictions as well as the final output chosen by the fusion module. We show mistakes in red, with the ground truth in parentheses.

the bias of `head` instances in VG, which are usually human heads, and the fact that `hat` instances typically co-occur with a `head`. Nevertheless, our commonsense model has the knowledge of object composition and knows `brids` typically have `heads` but not `hats`. Example (c) is an unusual case of `holding`, in terms of visual attributes such as arm pose. Hence, the perception model fails to predict `holding` correctly, while our commonsense model corrects that mistake by incorporating the affordance of `bottle`. Finally, in (d), the `person` is perceived `under` the `tower` due to the camera angle, but for the commonsense model that is unlikely due to intuitive physics. Hence, it corrects the mistake and the fusion module accepts that fix.

5.5 Summary

We presented the first method to learn visual commonsense automatically from a scene graph corpus. Our method learns structured commonsense patterns, rather than simple co-occurrence statistics, through a novel self-supervised training strategy. Our unique way of augmenting trans-

formers with local attention heads significantly outperforms transformers, as well as widely used graph-based models such as graph convolutional nets. Furthermore, we proposed a novel architecture for scene graph generation, which consists of two individual models, perception and commonsense, which are trained differently, and can complement each other under uncertainty, improving the overall robustness. To this end, we proposed a fusion mechanism to combine the output of those two models based on their confidences, and showed our model correctly determines when to trust its perception and when to fall back on its commonsense. Experiments show the effectiveness of our method for scene graph generation, and encourage future work to apply the same methodology on other computer vision tasks.

Chapter 6: Extending Graph-Based Representations to Multimedia Domain

In this chapter, we extend our earlier work on Visual Semantic Parsing (VSP, Chapter 3) to the domain of multimedia, where the input data includes not only visual content, but also text. To this end, we introduce a new task, **M**ulti**M**edia **E**vent **E**xtraction (M^2E^2), which aims to extract events and their arguments from multimedia documents. This results in a structured representation where nodes represent events and arguments, and edges represent the roles arguments play in each event, which closely resembles the structure of predicates and entities in VSP. Nevertheless, there are two key distinctions between VSP and M^2E^2due to the unique challenges of the multimedia domain: Firstly, in M^2E^2, each node may come from either image or text, which requires a seamless integration of information across modalities. Secondly, due to the more abstract semantic content of text compared to images, M^2E^2targets a higher-level ontology than VSP, where predicates correspond to high-level news events such as political conflicts, rather than low-level physical interactions such as holding or throwing an object.

In order to address the task of M^2E^2, we propose a novel method, **W**eakly **A**ligned **S**tructured **E**mbedding (**WASE**), which takes a mixture of visual and textual data as input and encodes both types of data into a unified semantic embedding space, while maintaining the inherent structure of entities and events. More specifically, WASE consists of a vision branch and a language branch, which extract a graph from the input image or sentence, where each node in each graph is represented as a vector in a shared embedding space. This way, a set of modality-agnostic classifiers can take each element (node or edge) of their input and classify their event and argument role, no matter which modality the input comes from. This seamless integration of modalities is enabled by the structured common space that is established through a novel weakly supervised, multitask training stragety. Our method is the first to employ graph-based representations for a multimedia understanding task, and the only available alternatives are either single-modality event extraction

models, or multimedia models that learn unstructured embeddings. Through experiments on a newly proposed dataset, we show that compared to state-of-the-art multimedia unstructured representations, we achieve 8.3% and 5.0% absolute F-score gains on multimedia event extraction and argument role labeling, respectively. Moreover, we show that our multimedia common space results in 9.8% absolute F-score gain compared to vision-only baselines, even in purely visual event extraction where the input does not contain text.

Disclaimer: This is a joint work with Manling Li from the University of Illinois at Urbana-Champaign. Some aspects of this work such as dataset construction and textual event extraction are not claimed as contributions of this thesis, but are briefly explained here to provide a better context. Moreover, the shared event and argument classifiers and vision-language graph alignment were both implemented by Manling Li, along with Figures 6.1-6.7. Furthermore, the entire chapter including all images, figures, tables, equations, and text are based on a recently published collaborative work [112].

6.1 Introduction

In Chapter 3, we proposed a new framework for extracting semantic graphs from images in order to understand their content. Nevertheless, in many applications such as journalism, information is transferred through multimedia data, such as videos and web pages that contain text and images. In those scenarios, knowledge is often distributed between visual and verbal forms in a holistic and entangled manner. This means a single news event that involves multiple entities (event arguments) may not be comprehensively described or shown in either of the modalities. More specifically, we randomly collected 100 multimedia news articles from the Voice of America (VOA), manually inspected them, and found that 33% of images in the articles contain visual objects that serve as event arguments and are not mentioned in the text. Considering Figure 6.1 as an example, the text includes information about a *Movement.Transport* event, along with its *Agent* and *Person* arguments, but only the image contains its *Vehicle* argument, even though the event is mentioned in text.

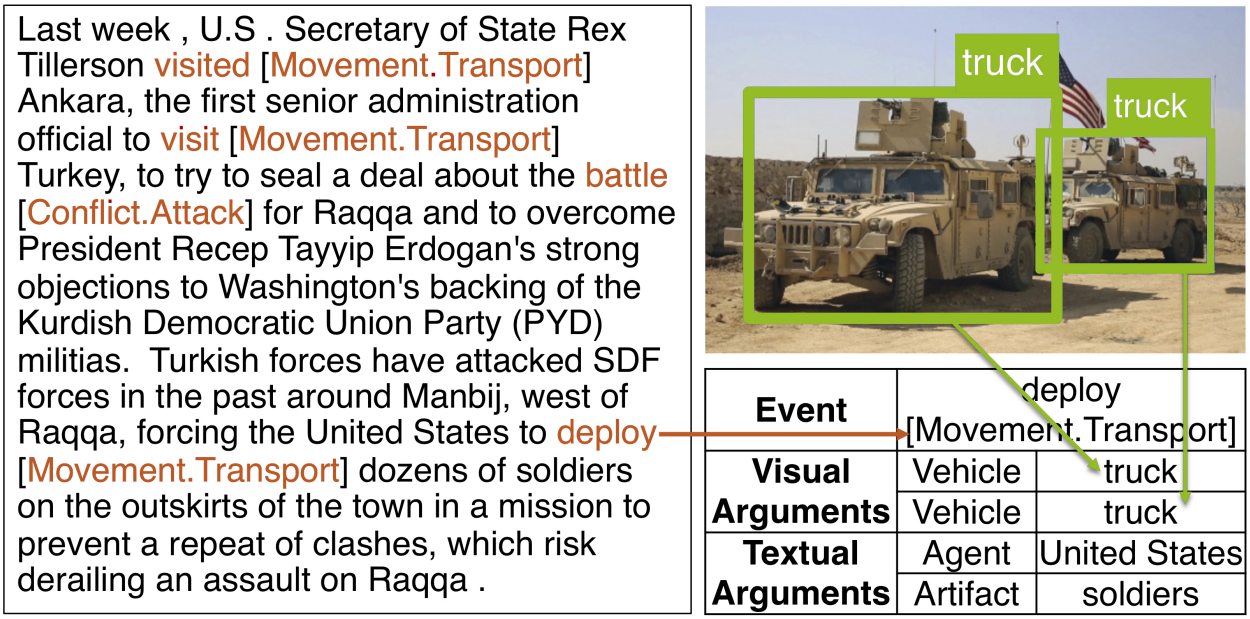


Figure 6.1: An example of our proposed task, Multimedia Event Extraction (M^2E^2). An event mention and some event arguments (*Agent* and *Person*) are mentioned in text, while the vehicle arguments only appear in the image.

Nevertheless, event extraction is independently studied in Computer Vision (CV) [42] and Natural Language Processing (NLP) [22], with major differences in task definition, data domain, methodology, and terminology. Therefore, it is essential to develop information extraction tools to understand multimedia data, which is an unexplored area of research. This is challenging due to the inherent semantic gap between verbal and visual information. Vision and language research has focused on distinct ontologies that were developed for each modality in isolation, and task formulations that were created for each modality separately. To enable unified multimedia understanding, it is crucial to first create a unified ontology and task formulation, and then to develop a single model that can extract the same type of knowledge representation from both modalities in an integrated and complementary manner.

Accordingly, we propose **M**ulti**M**edia **E**vent **E**xtraction (**M^2E^2**), a new task that aims to jointly extract events and arguments from multiple modalities. To this end, we extend the previously proposed formulation of VSP (Chapter 3) to the multimedia domain by integrating it with the task of Event Extraction (EE) in NLP [22]. In both tasks, the goal is to extract a set of events (predicates)

and arguments (entities), as well as the role the arguments play in each event. In VSP, entities and predicates are extracted from an image, while in EE, they are extracted from text. In M^2E^2, the goal is to extract the same kind of structure from multimedia data, which means each entity or event can come from either image, text, or both. Therefore, the graph extraction model should be agnostic about input modality. We achieve this by exploiting recent advances in multi-modality common space models.

M^2E^2can be seen as an extension of VSP (Chapter 3), since the output graphs have a similar structure. However, due to the semantic gap between visual and textual information, M^2E^2graphs are at a higher level of semantic abstraction compared to VSP graphs. More specifically, while *predicates* in VSP focus on simple visual interactions such as `holding` or `throwing`, their corresponding *events* in M^2E^2focus on higher-level newsworthy events such as `protest` or `attack`. Nevertheless, they both can be seen as events or interactions that involve entities through a set of semantic roles. Furthermore, since $M^2E^2$is a generalization of VSP, we design the visual branch of our model inspired by VSPNET (Chapter 3), as we elaborate in Section 6.4.1.

To address M^2E^2, We propose a novel method named **W**eakly **A**ligned **S**tructured **E**mbedding (**WASE**). The key idea of WASE is to learn a two-stream architecture where each stream extracts a structured representation from a single modality, and the output of both streams are within a common embedding space. Since the goal is to extract a semantic graph, we want the common space embedding to contain a structure, which has not been addressed by existing multimedia representation methods [113, 114, 115]. More specifically, given a multimedia document consisting of multiple sentences and images, we represent each image or sentence as a graph, where each node represents an event or entity and each edge represents an argument role. The node embeddings are represented in a multimedia common semantic space, where each two node with the same real-world meaning are close in that space, even if they come from different modalities. This enables us to jointly classify events and argument roles from both modalities, using a set of shared, modality-agnostic classifiers. WASE can also be seen as an extension of VSPNet to multimedia settings, because it features a similar entity-predicate message propagation and role-driven attention in its

visual branch, but also includes a language parsing branch which produces a similar representation.

Training any neural network for $M^2E^2$is challenging due to the lack of multimedia event argument annotations, which are costly to obtain due to the annotation complexity. Therefore, we propose a weakly supervised framework, which takes advantage of existing single-modality corpora to separately learn visual and textual event extraction, and uses an image-caption dataset to align the embedding space across modalities. After training WASE using the proposed approach, we evaluate it on a newly proposed dataset with M^2E^2annotations. Compared to the state-of-the-art single-modality methods and multimedia flat representations, our method significantly outperforms on both event extraction and argument role labeling tasks in all settings.

In summary, this chapter makes the following contributions:

- We propose a new task, MultiMedia Event Extraction, which extends our earlier work on structured image understanding to the multimedia domain.
- We propose a novel deep neural network architecture, which takes an image or sentence as input and extracts a unified structured representation from either modality, which is used to recognize events and argument roles across modalities
- We propose a joint cross-media training framework which uses weak supervision to learn event and argument role extraction from both modalities, while aligning the two modalities through image-caption pairs.

6.2 Related Work

6.2.1 Event Extraction

Event extraction has been extensively studied in the NLP literature for the general news domain [116, 117, 118, 119, 120, 121, 122, 123, 124, 125, 126, 127, 128, 22]. State-of-the-art methods typically use contextualized models to represent words or phrases in a given sentence, classify each to detect potential trigger words and recognize their event type, and classify each word paired with each event trigger to recognize their potential argument roles. Although event

extraction is a purely language-based task, multimedia features have been proven to effectively improve performance [129].

While “events" in NLP usually refer to complex events that involve multiple entities in a large span of time and space (e.g. protest), that is not often the case in computer vision. The majority of CV research on video understanding has been around atomic single-entity human actions (e.g. jumping) [130] or simple activities that may involve other non-human objects but within confined time and space boundaries (e.g. washing dishes) [131]. There is also progress towards more complex visual events (e.g. birthday party) [132], but those works consider each event as a singular concept and overlook the importance of structure, which is essential to understand the interactions between event arguments. There are a few methods that aim to localize the agent [130, 133, 134], or classify the recipient [135, 40, 136] of events, but neither detects the complete set of arguments for an event.

The most similar to our work is Situation Recognition (SR) [42, 44] which predicts an event and multiple arguments from an input image, but does not localize the arguments. Moreover, Silberer and Pinkal [137] redefine the problem of visual argument role labeling with event types and bounding boxes as input. Nevertheless, there is no existing solution for extracting events and arguments from multimedia data (both text and image) jointly. Different from prior work, we extend the problem scope to include event identification and coreference, and further advance argument localization by proposing an attention framework which does not require bounding boxes for training nor testing.

6.2.2 Multimedia Representation

There is a rich literature on multimedia representation learning, although not particularly studied for understanding events. Most multimedia representation research has been focused on simple applications such as image retrieval using natural language queries [138, 139, 140]. In such cases, each image or sentence can be represented holistically, using a single embedding vector. There are also works that represent each image or sentence as a set of vectors rather than one, usually

corresponding to objects and words respectively [141]. Those works are not only useful for image retrieval, but also a more fine-grained localization of referring expressions in text [142, 143, 144, 145]. More recently, there are new approaches that represent an image-sentence pair jointly, through several layers of cross-modality attention heads [1, 146, 147, 148]. Those models are more capable at jointly understanding multimedia content, and have been applied to more complex tasks such as visual question answering.

Nevertheless, existing works on multimedia representation ignore the inherent structure of semantic content, which has been shown essential for a variety of tasks in both vision and NLP, including event extraction [126]. UniVSE [113] incorporates entity attributes and relations into cross-media alignment, but does not capture graph-level structures of images or text, and uses the learned representation only for image retrieval. We create the first multimedia representation method where both image and text are represented as a graph, where each node is in a common space, and we show this architecture is more effective than alternative unstructured multimedia embedding methods.

6.3 Task Definition

In this section, we clearly define the new task of Multimedia Event Detection. The input to this task is a multimedia document, which consists of a set of images $\mathcal{M} = \{m_1, m_2, \dots\}$ and a set of sentences $\mathcal{S} = \{s_1, s_2, \dots\}$. Each sentence s can be represented as a sequence of tokens $s = (w_1, w_2, \dots)$, where w_i is a token from the document vocabulary $\mathcal{W}$. The input also includes a set of entities $\mathcal{T} = \{t_1, t_2, \dots\}$ extracted from the document text. An entity is an individually unique object in the real world, such as a person, an organization, a facility, a location, a geopolitical entity, a weapon, or a vehicle. The objective of $M^2E^2$is twofold:

Event Extraction: Given a multimedia document, extract a set of event mentions, where each event mention e has a type y_e and is grounded on a text trigger word w or an image m or both, i.e.,

$$e = (y_e, \{w, m\}). \tag{6.1}$$

Note that for an event, w and m can both exist, which means the visual event mention and the textual event mention refer to the same event. For example in Figure 6.1, *deploy* indicates the same *Movement.Transport* event as the image. We consider the event e as **text-only** event if it only has textual mention w, and as **image-only** event if it only contains visual mention m, and as **multimedia** event if both w and m exist.

Argument Extraction: The second task is to extract a set of arguments of event mention e. Each argument a has an argument role type y_a, and is grounded on a text entity t or an image object o (represented as a bounding box), or both,

$$a = (y_a, \{t, o\}) \,. \tag{6.2}$$

The arguments of visual and textual event mentions are merged if they refer to the same real-world event, as shown in Figure 6.1.

6.4 Method

In this section, we introduce our new model, **W**eakly **A**ligned **S**tructured **E**mbedding, which takes multimedia input and creates a structure consisting of events and arguments. We assume the input to our model is an image or a sentence or both. This is different from the definition of our task, M^2E^2, which assumes the input has a set of images and sentences. Hence, we augment WASE with preprocessing and postprocessing steps to split the input into image-sentence pairs and aggregate the output of WASE to build document-level structures. Those steps are explained in section 6.4.5 after presenting the core architecture of WASE.

WASE consists of a vision branch and a language branch, which take an image or sentence as input, respectively, and extract a graph-based representation in a common semantic space. Then an event classifier and an argument classifier take those graphs and classify each element to create a symbolic graph with event types and argument roles. Each branch can be trained separately using an annotated dataset in the corresponding modality. However, we choose to share the classifiers

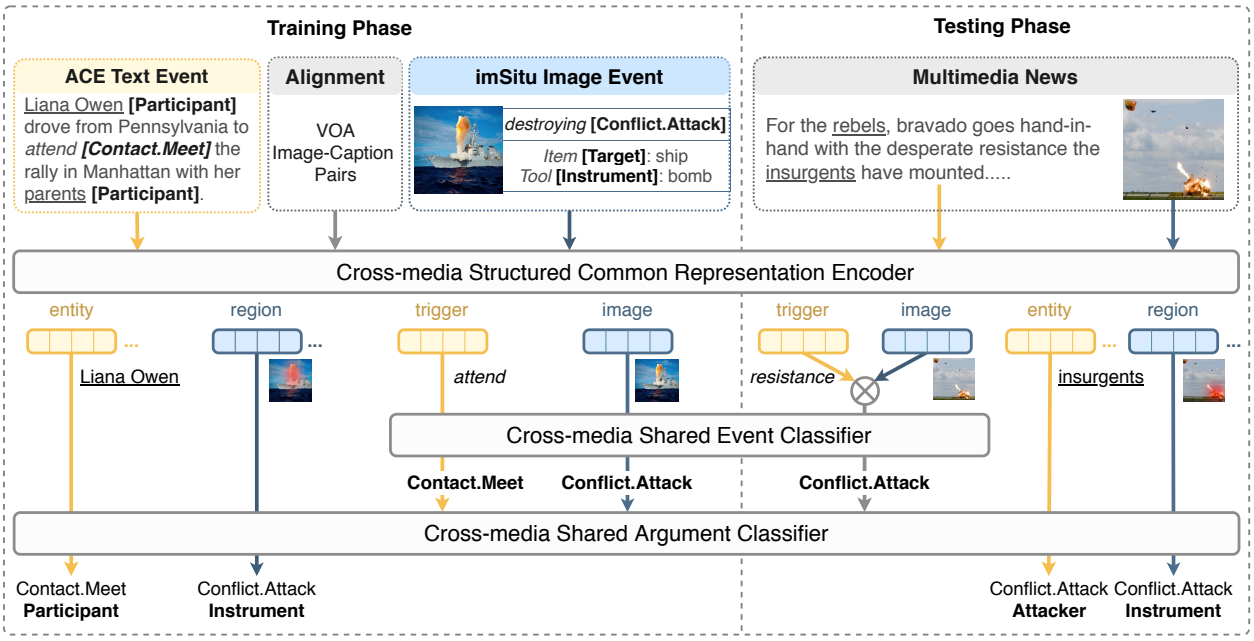


Figure 6.2: An overview of our proposed training paradigm. During training (left), we simultaneously learn three tasks, which not only learn text-based and image-based event and argument extraction, but also learn to do so via a shared set of classifiers that are agnostic about the input modality. During test (right), our multimedia shared embedding can be used to jointly extract events and arguments from multimedia articles.

across modalities and train the two branches jointly, even though training sources are distinct and not multimedia. To further promote the formation of a uniform common space with identical distribution of modalities, we devise an auxiliary task that aligns the outputs of the two branches using paired image-caption data. Note that in test time, there is no need for input sentences to be captions.

In the rest of this section, we elaborate how we adopt the previously proposed VSPNet (Chapter 3) to form the visual branch of WASE. Then we briefly describe the language branch, and then the shared classifiers. Next, we elaborate the training framework, as well as the inference procedure. Figure 6.3 illustrates the architecture of our proposed model and Figure 6.2 shows an overview of our training and inference mechanism.

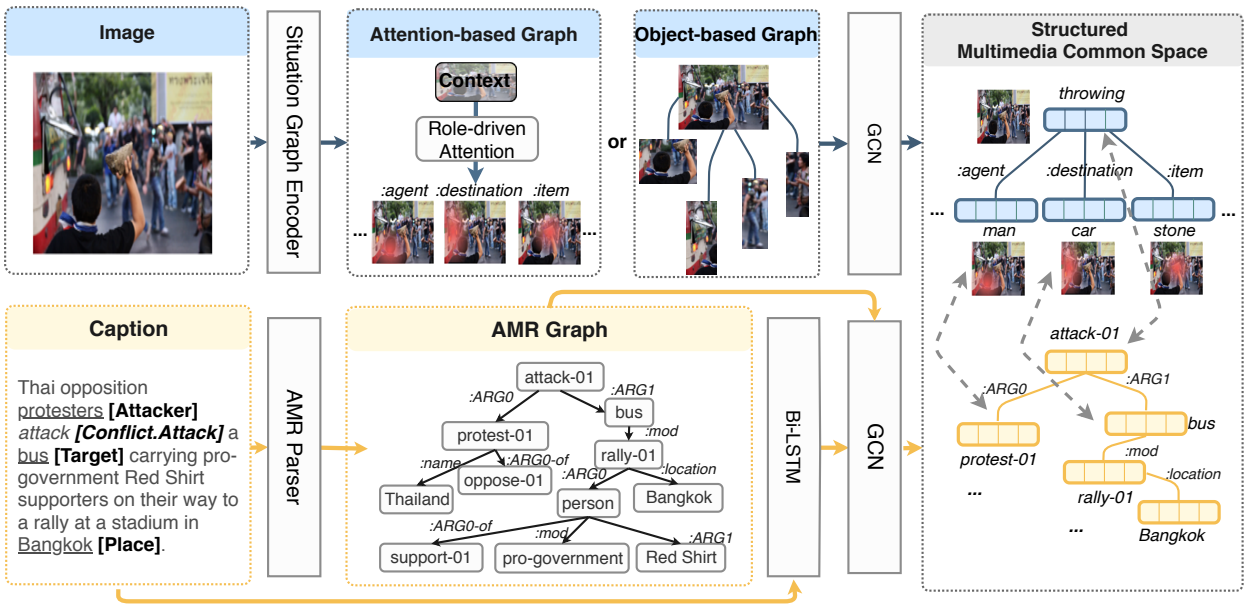


Figure 6.3: The two-stream architecture of our **W**eakly **A**ligned **S**tructured **E**mbeddingmodel. A vision branch takes an input image and extracts a graph with contextualized node representations using either an attention-based or object-based graph initialization mechanism. In parallel, a language branch extracts a similar representation based on an AMR graph. The two output graphs are projected onto a common semantic space where nodes that convey the same meaning are close to each other, even if they come from different modalities. Red pixels depict the attention heatmaps of our attention-based model.

6.4.1 Structured Visual Embedding Branch

Similar to VSPNet, we aim to extract entities and predicates from each image, as well as potential semantic roles, and represent each node via a contextualized embedding. However, in this task, events can be seen as a high-level form of predicates, which typically involve more arguments in a broader span of time and space. For instance, a `protest` event in $M^2E^2$is similar to the predicate `hold` in VSP, because both represent a situation that can happen in real world. However, `hold` only involves an agent and an object that is being held, and also possibly an instrument such as `hand`, while a `protest` is a much more complex situation involving multiple agents doing different things, such as holding banners or shouting, in a much larger area and during a typically longer time. Although both `protest` and `hold` can be represented as a single verb in text, they appear much more differently in visual content. Hence, VSPNet may not directly work

on the news events domain.

To adapt VSPNet to higher-level events, we assume each image can only contain one event, and that event is covered by the entire image context, rather than a local region. This means we have a single predicate node in our graph, surrounded by a few entity nodes that comprise its arguments, essentially forming a star-shaped structure. This also aligns with the assumptions made by Situation Recognition [42]. Therefore, we call this star-shaped structure a *situation graph*, and train it on the imSitu dataset [42] in a weakly supervised fashion to perform the task of situation recognition.

More specifically, the central node represents a verb v (e.g., *destroying*), and the neighbor nodes are arguments represented as $\{(n, r)\}$, where n is a noun (e.g., *ship*) derived from WordNet synsets [81] to indicate the entity type, and r indicates the role (e.g., *item*) played by the entity in the event, based on FrameNet [149]. Due to the simplified structure of our situation graph, we found it sufficient to use a fixed graphical structure for context propagation without iteratively refining the edges as in VSPNet. We explore two different ways to generate the graph, and discuss their trade-offs in Section 6.5.2.

(1) Object-based Graph: Similar to VSPNet, we use a Faster R-CNN backbone trained on Open Images [72] (which has 600 object classes) to extract initial bounding boxes. We use a VGG-16 CNN [150] to extract visual features from the entire image $\boldsymbol{m}$, and use that to represent the event node. We use another VGG-16 backbone (identically initialized, but independently trained) to encode the bounding boxes $\{\boldsymbol{o}_i\}$. As an auxiliary means to train each backbone, we apply a fully connected network to predict a verb embedding from $\boldsymbol{m}$ and another fully connected to predict a noun embedding for each $\boldsymbol{o}_i$.

$$\hat{\boldsymbol{m}} = \text{MLP}_\text{m}(\boldsymbol{m}) \ , \ \hat{\boldsymbol{o}}_i = \text{MLP}_\text{o}(\boldsymbol{o}_i). \tag{6.3}$$

We compare the predicted verb embedding to all verbs v in the imSitu taxonomy in order to classify the verb, and similarly compare each predicted noun embedding to all imSitu nouns n which results

in probability distributions:

$$P(v|m) = \frac{\exp\left(\hat{\boldsymbol{m}}\boldsymbol{v}\right)}{\sum_{v'}\exp\left(\hat{\boldsymbol{m}}\boldsymbol{v'}\right)},$$
$$P(n|o_i) = \frac{\exp(\hat{\boldsymbol{o}}_i\boldsymbol{n})}{\sum_{n'}\exp(\hat{\boldsymbol{o}}_i\boldsymbol{n'})}, \tag{6.4}$$

where $\boldsymbol{v}$ and $\boldsymbol{n}$ are word embeddings initialized with GloVE [74]. We use another fully connected network with one hidden layer followed by Softmax (σ) to classify role r_i for each object o_i:

$$P(r_i|o_i) = \sigma\big(\mathrm{MLP_r}(\hat{\boldsymbol{o}}_i)\big). \tag{6.5}$$

Given verb v^* and role-noun (r_i^*, n_i^*) annotations for an image (from the imSitu corpus), we define the situation loss functions:

$$\mathcal{L}_v = -\log P(v^*|m),$$
$$\mathcal{L}_r = -\log(P(r_i^*|o_i) + P(n_i^*|o_i)). \tag{6.6}$$

(2) Attention-based Graph: State-of-the-art object detection methods only cover a limited set of object types, such as the 600 classes in Open Images. Many salient and newsworthy objects such as *bomb*, *stone* and *stretcher* are not covered in these ontologies. Hence, we propose an open-vocabulary alternative to the object-based graph construction model. To this end, we bypass the object proposal network and directly use the image feature map as an input to the role-driven attention introduced in Chapter 3. More specifically, we use a VGG-16 CNN to extract a 7×7 convolutional feature map for each image m, which can be regarded as attention *keys* $\boldsymbol{k}_i$ for 7×7 local regions. Next, for each role r defined in the situation recognition ontology (e.g., *agent*), we build an attention *query* vector $\boldsymbol{q}_r$ by concatenating role embedding $\boldsymbol{r}$ with the image feature $\boldsymbol{m}$ as context and apply a fully connected layer:

$$\boldsymbol{q}_r = \boldsymbol{W}_q[\boldsymbol{r}; \boldsymbol{m}] + \boldsymbol{b}_q. \tag{6.7}$$

Then, we compute the dot product of each query with all keys, followed by Softmax, which forms

a heatmap $\boldsymbol{h}$ on the image, i.e.,

$$h_i = \frac{\exp(\boldsymbol{q}_r \boldsymbol{k}_i)}{\sum_{j \in 7\times 7} \exp(\boldsymbol{q}_r \boldsymbol{k}_j)}. \tag{6.8}$$

We use the heatmap to obtain a weighted average of the feature map to represent the argument o_r of each role r in the visual space:

$$\boldsymbol{o}_r = \sum_i h_i \boldsymbol{m}_i. \tag{6.9}$$

Similar to the object-based model, we embed $\boldsymbol{o}_r$ to $\hat{\boldsymbol{o}_r}$, compare it to the imSitu noun embeddings to define a distribution, and define a classification loss function. The verb embedding $\hat{\boldsymbol{m}}$ and the verb prediction probability $P(v|m)$ and loss are defined in the same way as in the object-based method.

Once we initialize the situation graph using either the object-based or attention-based mechanisms and pretrain the backbones on the imSitu dataset [42], we complete the visual embedding branch by applying a Graph Convolutional Network (GCN) [47] to propagate the context similar to VSPNet:

$$\boldsymbol{x}_i^{(k+1)} = f\Big(\sum_{j \in \mathcal{N}(i)} g_{ij}^{(k)} (\boldsymbol{W}_{E(i,j)}^{(k)} \boldsymbol{x}_j^{(k)} + \boldsymbol{b}_{E(i,j)}^{(k)})\Big), \tag{6.10}$$

where $\boldsymbol{x}_i^{(k)}$ is the embedding of node i at layer k, $\mathcal{N}(i)$ is the set of nodes adjacent to $\boldsymbol{x}_i$, $E(i,j)$ is the edge type (semantic role) between $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$, g_{ij} is a message aggregation gate, and f is the Sigmoid function. The set of nodes $\{\boldsymbol{x}_i\}$ consist of the event node $\boldsymbol{m}$ as well as entity nodes $\boldsymbol{o}_j$. We take the hidden states of the last GCN layer for each node as the common-space representation $\boldsymbol{m}^{\mathbb{C}}$ and $\boldsymbol{o}_i^{\mathbb{C}}$, where $\mathbb{C}$ stands for the common (multimedia) embedding space.

6.4.2 Structured Language Embedding Branch

The language branch of WASE is outside the scope of this thesis, but we briefly describe it for completeness. For more details refer to [112]. This branch takes a sentence as input and extracts a graph-based embedding based on adopt Abstract Meaning Representation (AMR) [151]. AMR is a suitable choice both because it is a comprehensive semantic representation, and also because it resembles our visual structure in that both graphs consist of edges labeled with semantic

roles that connect predicates to their arguments. To encode each text sentence, we run the CAMR parser [152, 153, 154] to generate an AMR graph, based on the named entity recognition and part-of-speech (POS) tagging results from Stanford CoreNLP [155]. To represent each word w in a sentence s, we concatenate its pre-trained GloVe word embedding [74], POS embedding, entity type embedding and position embedding. We then input the word sequence to a bi-directional long short term memory (Bi-LSTM) [156] network to encode the word order and get the representation of each word $\boldsymbol{w}$.

Similar to the vision branch, we apply a Graph Convolutional Network (GCN) [47] on the AMR graph to propagate context following [126]. More specifically:

$$\boldsymbol{w}_i^{(k+1)} = f(\sum_{j \in \mathcal{N}(i)} g_{ij}^{(k)} (\boldsymbol{W}_{E(i,j)}^{(k)} \boldsymbol{w}_j^{(k)} + \boldsymbol{b}_{E(i,j)}^{(k)})), \tag{6.11}$$

where $\mathcal{N}(i)$ is the neighbour nodes of w_i in the AMR graph, $E(i, j)$ is the edge type between w_i and w_j, g_{ij} is the gate following [126], k represents GCN layer number, and f is the Sigmoid function. We take the hidden states of the last GCN layer for each word as the common-space representation $\boldsymbol{w}^{\mathbb{C}}$. For each entity t, we obtain its representation $\boldsymbol{t}^{\mathbb{C}}$ by averaging the embeddings of its tokens.

6.4.3 Cross-Media Shared Classifiers

The goal of M^{2}E^{2}is to create a unified semantic representation that abstracts away from modality. Hence, it is essential to create an extraction model that can ground each node of the output graph on any modality in a seamless manner. To this end, one of our key design principles is to share the final classifiers among modalities. This means each event is classified into an event type y_e[1] no matter if the input is the visual event node of the visual graph $\boldsymbol{m}^{\mathbb{C}}$ or a potential trigger word from the text graph $\boldsymbol{w}^{\mathbb{C}}$. Similarly, each argument is classified into a role y_a, no matter if the input is a visual object $\boldsymbol{o}^{\mathbb{C}}$ or text entity $\boldsymbol{t}^{\mathbb{C}}$.

[1]We use BIO tag schema to decide trigger word boundary, i.e., adding prefix *B-* to the type label to mark the beginning of a trigger, *I-* for inside, and *O* for none.

We define the class-agnostic classifiers that are applied on the text graph as follows:

$$
\begin{aligned}
P(y_e|w) &= \frac{\exp\left(\boldsymbol{W}_e \boldsymbol{w}^{\mathbb{C}} + \boldsymbol{b}_e\right)}{\sum_{e'} \exp\left(\boldsymbol{W}_{e'} \boldsymbol{w}^{\mathbb{C}} + \boldsymbol{b}_{e'}\right)}, \\
P(y_a|t) &= \frac{\exp(\boldsymbol{W}_a [\boldsymbol{t}^{\mathbb{C}}; \boldsymbol{w}^{\mathbb{C}}] + \boldsymbol{b}_a)}{\sum_{a'} \exp(\boldsymbol{W}_{a'} [\boldsymbol{t}^{\mathbb{C}}; \boldsymbol{w}^{\mathbb{C}}] + \boldsymbol{b}_{a'})},
\end{aligned}
\tag{6.12}
$$

and similarly applied on the visual graph as follows:

$$
\begin{aligned}
P(y_e|m) &= \frac{\exp(\boldsymbol{W}_e \boldsymbol{m}^{\mathbb{C}} + \boldsymbol{b}_e)}{\sum_{e'} \exp(\boldsymbol{W}_{e'} \boldsymbol{m}^{\mathbb{C}} + \boldsymbol{b}_{e'})}, \\
P(y_a|o) &= \frac{\exp(\boldsymbol{W}_a [\boldsymbol{o}^{\mathbb{C}}; \boldsymbol{m}^{\mathbb{C}}] + \boldsymbol{b}_a)}{\sum_{a'} \exp(\boldsymbol{W}_{a'} [\boldsymbol{o}^{\mathbb{C}}; \boldsymbol{m}^{\mathbb{C}}] + \boldsymbol{b}_{a'})}.
\end{aligned}
\tag{6.13}
$$

6.4.4 Multimedia Joint Training

In order to make the event and argument classifier shared across modalities, the image and text graph should be represented within the same semantic space. However, it is extremely costly to obtain parallel multimedia event and argument annotation. Hence, we use event and argument annotations in separate modalities (i.e., ACE and imSitu datasets for text and vision respectively) to train the model. Although it is possible to simultaneously optimize both tasks using shared classifier weights, this does not guarantee a coherent semantic space at the input of the classifier, which prevents cross-media aggregation or generalization ability. Therefore, we introduce an additional auxiliary task to ensure the two modalities align well on the shared embedding space.

More specifically, we use a dataset of image-caption pairs from the news domain to extract the visual and text embedding graphs and ensure the corresponding nodes of those graphs are relatively close compared to non-matching image-caption pairs. Since there is no ground truth alignment between the image nodes and caption nodes, we devise a weakly supervised alignment

technique based on a soft cross-media attention mechanism. Concretely,

$$\alpha_{ij} = \frac{\exp\left(\boldsymbol{w}_{\boldsymbol{i}}^{\mathbb{C}} \boldsymbol{x}_{\boldsymbol{j}}^{\mathbb{C}}\right)}{\sum_{j'} \exp\left(\boldsymbol{w}_{\boldsymbol{i}}^{\mathbb{C}} \boldsymbol{x}_{\boldsymbol{j'}}^{\mathbb{C}}\right)}, \beta_{ji} = \frac{\exp\left(\boldsymbol{w}_{\boldsymbol{i}}^{\mathbb{C}} \boldsymbol{x}_{\boldsymbol{j}}^{\mathbb{C}}\right)}{\sum_{i'} \exp\left(\boldsymbol{w}_{\boldsymbol{i'}}^{\mathbb{C}} \boldsymbol{x}_{\boldsymbol{j}}^{\mathbb{C}}\right)}, \tag{6.14}$$

where $\boldsymbol{w}_i$ indicates the i^{th} word in caption sentence s and $\boldsymbol{x}_j$ represents the j^{th} node of the image graph, including objects $\boldsymbol{o}$ and the event node $\boldsymbol{m}$. α and β represent attention weights from caption nodes to image nodes and vice versa. We use these attention weights to compute the attended visual embedding for each caption node and vice versa:

$$\boldsymbol{w}_i' = \sum_j \alpha_{ij} \boldsymbol{x}_j^{\mathbb{C}} \ , \ \boldsymbol{x}_j' = \sum_i \beta_{ji} \boldsymbol{w}_i^{\mathbb{C}}. \tag{6.15}$$

$\boldsymbol{w}_i'$ can be seen as a reconstruction of $\boldsymbol{w}_i$ using visual nodes, and we want to ensure that the visual embeddings can reconstruct $\boldsymbol{w}_i$ as closely as possible. Similarly, we want text embeddings to reconstruct $\boldsymbol{x}_j$ accurately. Hence, we define the alignment loss as the Euclidean distance between each node and its reconstruction:

$$\langle s, m \rangle = \sum_i ||\boldsymbol{w}_i - \boldsymbol{w}_i'||_2^2 + \sum_j ||\boldsymbol{x}_j - \boldsymbol{x}_j'||_2^2. \tag{6.16}$$

We use a triplet loss to pull relevant image-caption pairs close while pushing irrelevant ones apart:

$$\mathcal{L}_c = \max(0, 1 + \langle s, m \rangle - \langle s, m^- \rangle), \tag{6.17}$$

where m^- is a randomly sampled negative image that does not match s.

The common space enables the event and argument classifiers to share weights across modalities, and be trained jointly on the ACE and imSitu datasets, by minimizing the following objective functions:

$$\begin{aligned} \mathcal{L}_e &= -\sum_w \log P(y_e|w) - \sum_m \log P(y_e|m), \\ \mathcal{L}_a &= -\sum_t \log P(y_a|t) - \sum_o \log P(y_a|o), \end{aligned} \tag{6.18}$$

All tasks are jointly optimized:

$$\mathcal{L} = \mathcal{L}_v + \mathcal{L}_r + \mathcal{L}_e + \mathcal{L}_a + \mathcal{L}_c \tag{6.19}$$

Optimizing $\mathcal{L}$ entails learning three tasks, each using a separate dataset, namely ACE [157], imSitu [42], and the image-caption pairs in our VOA dataset as described in section 6.5.1. To this end, in each training step, we sample a batch from each of those datasets to calculate the corresponding loss and its gradients, and update all parameters simultaneously using the sum of all gradients. The text event extraction task ($\mathcal{L}_e + \mathcal{L}_a$) updates the shared event and role classifiers, as well as the parameters of the language branch, including GCN, LSTM, and word embeddings. The visual event extraction task ($\mathcal{L}_v + \mathcal{L}_e$) updates the same shared classifiers as well as the parameters of the vision branch, including GCN, attention, and the CNN backbones. The cross-media alignment task ($\mathcal{L}_c$) updates the parameters of the vision branch as well as the language branch. Collectively, these three tasks lead the vision and language branches to extract unified representations that contain sufficient information for recognizing events and their arguments in a seamless manner.

6.4.5 Multimedia Joint Inference

$M^2E^2$is a challenging task because news events are defined as a composition of complex interactions between a variety of entities that may appear in different modalities. Hence, although we rely on single-modality data sources in training, we prefer multimedia joint inference in test time, in order to incorporate the information from both modalities before making a decision about each event. To this end, during inference, we aggregate the embedding of each node of the input sentence s with the aligned embedding of nodes from the paired image via weighted averaging:

$$\boldsymbol{w}_i'' = (1 - \gamma)\boldsymbol{w}_i + \gamma \boldsymbol{w}_i', \tag{6.20}$$

where $\gamma = \exp(-\langle s, m \rangle)$ and $\boldsymbol{w}_i'$ is derived from m using Equation 6.15. We use $\boldsymbol{w}_i''$ to classify each word into an event type and to classify each entity into a role with multimedia classifiers in

Equation 6.12. Similarly, for the image m, we compute the aggregated multimedia features $\boldsymbol{m}''$ and $\boldsymbol{o}_i''$, and feed into the shared classifiers (Equation 6.13) to predict visual event and argument roles. Finally, we use the similarity of the image-sentence pair $\langle s, m \rangle$ to determine whether the events extracted from the two modalities are coreferential. This is the key to extract events that have arguments in both modalities. It is also possible that two arguments from the two modalities are coreferential, which can be merged using cross-media entity coreference resolution [145], but that is outside the scope of this work.

In the M^{2}E^{2}task, the input is a multimedia document with sentences $S = \{s_1, s_2, \dots\}$ and images $M = \{m_1, m_2, \dots, \}$. This is while our proposed WASE model takes a single image and caption at a time. To enable document-level extraction, we first generate the structured common embedding for each sentence and each image, and then compute pairwise similarities $\langle s, m \rangle$ between all image-caption pairs. We pair each sentence s with the closest image m, and feed each pair to the model for joint classification. Each pair results in a small graph with a handful of events, and we finally get the union of all graphs to represent the document. In order to create a coherent graph for the entire document, an additional coreference resolution step is required, which is outside the scope of this work.

6.5 Experiments

6.5.1 Evaluation Setting

Dataset We conduct experiments on the M^{2}E^{2}dataset, which is created for this task and presented at [112]. M^{2}E^{2}consists of 245 selected articles from the Voice of America (VOA) website [2]. Those articles collectively contain 6,167 sentences and 1,014 images, all annotated for events and arguments. There are 1,297 annotated instances of events in text, as well as 1,965 arguments, while there are 391 events and 1,429 arguments annotated in images. Moreover, there are 309 events that are both mentioned in text and image, making the multimedia portion of the evaluation set, leaving 1,105 text-only and 188 image-only events. We separately evaluate on the multimedia and image-

[2] https://www.voanews.com/

only portions of this dataset to show the effectiveness of multimedia training on image-only data as well. Note that this dataset was only used for evaluation and not training. Table 6.1 shows the M^2E^2ontology and statistics.

Table 6.1: The taxonomy of event types and argument roles in M^2E^2, along with the frequency of each type in the (text|image) parts of the dataset.

Event Type	**Argument Role**
Movement.Transport (223|53)	Agent (46|64), Artifact (179|103), Vehicle (24|51), Destination (120|0), Origin (66|0)
Conflict.Attack (326|27)	Attacker (192|12), Target (207|19), Instrument (37|15), Place (121|0)
Conflict.Demonstrate (151|69)	Entity (102|184), Police (3|26), Instrument (0|118), Place (86|25)
Justice.ArrestJail (160|56)	Agent (64|119), Person (147|99), Instrument (0|11), Place (43|0)
Contact.PhoneWrite (33|37)	Entity (33|46), Instrument (0|43), Place (8|0)
Contact.Meet (127|79)	Participant (119|321), Place (68|0)
Life.Die (244|64)	Agent (39|0), Instrument (4|2), Victim (165|155), Place (54|0)
Transaction.TransferMoney (33|6)	Giver (19|3), Recipient (19|5), Money (0|8)

Evaluation Metrics We conduct evaluation on the image-only and multimedia event mentions in the M^2E^2dataset and adopt traditional event extraction measures, i.e., *Precision*, *Recall* and F_1 for events and argument roles separately. In the NLP literature [116, 119], an event mention is correct if its event type and trigger offsets match a reference trigger; and an event argument is correct if its event type, offsets, and role label match a reference argument. We make a similar definition for the visual domain: a visual event mention is correct if its event type and image ID match a reference visual event mention; and a visual event argument is correct if its event type, localization, and role label match a reference argument. A visual argument is correctly localized if the Intersection over Union (IoU) of the predicted bounding box with the ground truth bounding box is over 0.5. Subsequently, we define a multimedia event mention to be correct if its event type and trigger word/image match the reference. The arguments of multimedia events are either textual or visual arguments, and are evaluated accordingly.

Baselines We compare two variants of our WASE model with the object-based and attention-based graph initialization mechanisms, denoted as WASE_{obj} and WASE_{att} respectively. To show

the effectiveness of structured embedding, we include a baseline by removing the text and image GCNs from our model, which is denoted as Flat. The Flat baseline ignores edges and treats images and sentences as sets of vectors. We also compare to the state-of-the-art cross-media common representation model, Contrastive Visual Semantic Embedding VSE-C [158], by training it the same way as WASE. Moreover, we show the effectiveness of multimedia training by removing the vision branch and the language branch of our network, one at a time, denoted as $\text{WASE}^{\mathbb{T}}$ and $\text{WASE}^{\mathbb{I}}$ respectively.

6.5.2 Quantitative Results

Table 6.2: Presicion, recall, and F_1 scores of our method compared to various baselines on the M^2E^2dataset (%).

	Image-Only Evaluation						**Multimedia Evaluation**					
Method	**Event Mention**			**Argument Role**			**Event Mention**			**Argument Role**		
	P	*R*	F_1	*P*	*R*	F_1	*P*	*R*	F_1	*P*	*R*	F_1
$\text{WASE}^{\mathbb{T}}$	-	-	-	-	-	-	41.2	33.1	36.7	20.1	13.0	15.7
$\text{WASE}^{\mathbb{I}}_{\text{att}}$	29.7	61.9	40.1	9.1	10.2	9.6	28.3	23.0	25.4	2.9	6.1	3.8
$\text{WASE}^{\mathbb{I}}_{\text{obj}}$	28.6	59.2	38.7	13.3	9.8	11.2	26.1	22.4	24.1	4.7	5.0	4.9
VSE-C	30.3	48.9	26.4	5.6	6.1	5.7	33.3	48.2	39.3	11.1	14.9	12.8
Flat_{att}	27.1	57.3	36.7	4.3	8.9	5.8	33.9	59.8	42.2	12.9	17.6	14.9
Flat_{obj}	26.4	55.8	35.8	9.1	6.5	7.6	34.1	56.4	42.5	16.3	15.9	16.1
WASE_{att}	32.3	63.4	42.8	9.7	11.1	10.3	38.2	67.1	49.1	18.6	21.6	**19.9**
WASE_{obj}	43.1	59.2	**49.9**	14.5	10.1	**11.9**	43.0	62.1	**50.8**	19.5	18.9	19.2

As shown in Table 6.2, our complete methods (WASE_{att} and WASE_{obj}) outperform all baselines on all tasks in terms of F_1. The superiority over the flat baselines as well as VSE-C demonstrate the importance of structured representations for event understanding, which is effectively learned by WASE. The improvement over the text-only and vision-only variants of our method on the multimedia task prove that none of the modalities are sufficient for fully understanding multimedia news events. More interestingly, we observe an improvement over the vision-only baselines even in the image-only evaluation setting, which is counter-intuitive as a vision-only model is more directly optimized to perform that task, and training on text data should not directly improve visual

understanding. We believe this improvement is due to the fact that learning a multimedia common embedding space enables knowledge transfer and generalization ability across modalities.

WASE_{obj} and WASE_{att}, are both superior to the state of the art and each has its own advantages. WASE_{obj} predicts more accurate bounding boxes since it is based on a Faster R-CNN pretrained on bounding box annotations, resulting in a higher argument precision. While WASE_{att} achieves a higher argument recall as it is not limited by the predefined object classes of the Faster R-CNN.

Table 6.3: Presicion, recall, and F_1 scores on the cross-media event coreference task of the M^2E^2dataset.

Model	P (%)	R (%)	F_1 (%)
rule_based	10.1	100	18.2
VSE	31.2	74.5	44.0
Flat_{att}	33.1	73.5	45.6
Flat_{obj}	34.3	76.4	47.3
WASE_{att}	39.5	73.5	51.5
WASE_{obj}	40.1	75.4	52.4

To additionally show the quality of our multimedia common space, we evaluate cross-media event coreference using M^2E^2annotation. To this end, we feed all possible pairs of textual and visual event mentions that exist in the same document to our model separately, and use the image-sentence distance (Eq. 6.16) to determine how likely they correspond to the same event. We calculate *Precision*, *Recall* and F_1 for retrieving ground truth pairs. As shown in Table 6.3, WASE_{obj} outperforms all multimedia embedding models, as well as a rule-based baseline that simply matches events with the same type. This demonstrates the effectiveness of our weakly supervised cross-media alignment technique.

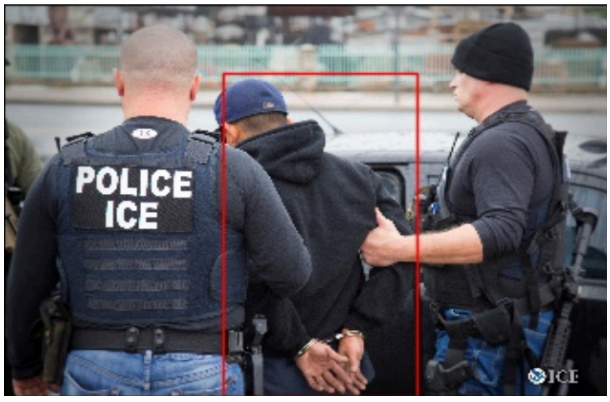
6.5.3 Qualitative Analysis

One key message of this chapter is that structured representations are essential for understanding complex situations such as news events. While Table 6.2 and Table 6.3 clearly prove this hypothesis, here we provide additional qualitative evidence. Figure 6.4 compares the predictions of our method with the flat embedding baseline. In these cases, the relevant position of objects

provide important cues for identifying the arguments. For instance, the fact that people are located on the truck means they are the artifacts of the transport event, and the man in the middle of two police officers is the target target of the arrest rather than its agent. Flat embeddings disregard the structure of the image and hence are unable to maintain these crucial information.

In Figure 6.5, we further showcase that visual information is sometimes not sufficient, or even misleading, as the image-only WASE classifies that image as a *Conflict.Demonstration*. This is while the sentences clearly describes the event as a celebration, which help our multimedia WASE model correct that mistake.

Flat	Event	Movement.Transport
	Role	Artifact = none
Ours	Event	Movement.Transport
	Role	Artifact = man

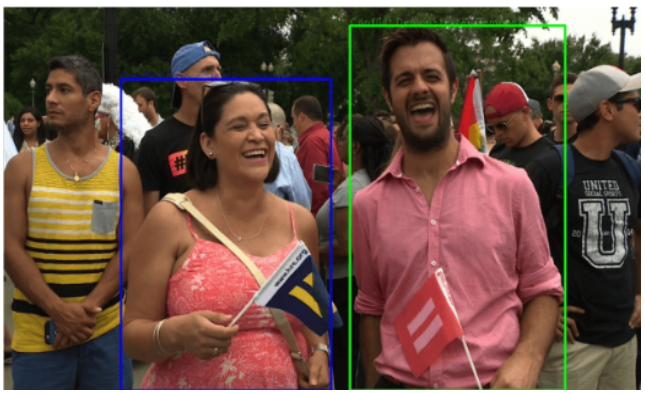


Flat	Event	Justice:ArrestJail
	Role	Agent = man
Ours	Event	Conflict.Attack
	Role	Entity = man

Figure 6.4: A qualitative comparison of our method's output with the flat embedding baseline.

One of the biggest challenges in $M^2E^2$is localizing arguments in images. Object-based models suffer from the limited object types, while attention-based models are not able to precisely localize objects, due to the lack of bounding box supervision. For instance in Figure 6.6, the *Entity* argument in the *Conflict.Demonstrate* event is correctly predicted as *troops*, but its localization is incorrect, while the *Place* argument is correctly localized as well. Furthermore, for arguments that are not clearly salient in the image, attention heatmaps tend to lose focus and cover the entire

Iraqi security forces *search* ***[Justice.Arrest]*** a civilian in the city of Mosul.



People celebrate Supreme Court ruling on Same Sex Marriage in front of the Supreme Court in Washington.

Figure 6.5: Examples of multimedia data facilitating accurate event extraction, when a single modality is not sufficient. Left: the image helps disambiguate the word "search" mentioned in text. Right: the text clearly describes the event while the image is not sufficient.

image, as shown in Figure 6.7.

Figure 6.6: An example of incorrect argument localization, while the argument entity type (troops) is correctly recognized.

6.6 Summary

In this chapter, we presented an extension of our visual semantic parsing paradigm to multimedia settings, where the input is not only an image, but a multimedia document containing images as well as text. To this end, we proposed a new task, Multimedia Event Extraction (M^2E^2), which unifies visual semantic parsing with the NLP task of event and argument extraction. M^2E^2provides a unified view of semantic graphs across modalities, which enables a seamless extraction of co-

Figure 6.7: An example of incorrect argument localization, due to the attention losing focus on small objects.

herent and inter-connected information from multimedia data. Based on that, we propose a novel neural architecture which integrates two modality-specific (vision and language) branches into a modality-agnostic structured semantic representation. Our model, **W**eakly **A**ligned **S**tructured **E**mbedding(WASE) extracts graph-based representations from images as well as sentences into a common embedding space, where a set of class-agnostic classifiers can identify events and argument roles no matter which modality each come from. This results in the first system capable of extracting events and their full argument set jointly from multimedia data. Through experiments on a newly collected dataset of multimedia news articles, we demonstrate the effectiveness of WASE in extracting events and argument roles, resulting in a new step towards understanding events in multimedia data.

Chapter 7: Extension to Open-Vocabulary Objects

In all previous chapters, we limited our models to closed-vocabulary settings, which is a ubiquitous convention in almost every area of computer vision. In other words, we always trained models to detect a predefined set of concepts (*e.g.* entity or predicate types), assuming sufficient supervision is available for every class. However, this limits the scalability of our work to other domains which may involve other concepts. In this chapter, we study an extension of our work to open-vocabulary settings, by introducing a new paradigm named open-vocabulary scene understanding. Specifically, we are inspired by the human ability to learn vision and language naturally without explicit supervision, and utilize that to perform various tasks. To this end, we revisit the task of object detection, which is a fundamental element of scene understanding, and an integral part of all SGG methods.

Object detectors require costly supervision to be trained, and learning more object categories typically requires proportionally more bounding box annotations. Weakly supervised and zero-shot learning techniques have been explored to scale object detectors to more categories with less supervision, but they have not been as successful and widely adopted as supervised models. In this chapter, we put forth a novel formulation of the object detection problem, namely open-vocabulary object detection, which is more general, more practical, and more effective than weakly supervised and zero-shot approaches. We propose a new method to train object detectors using bounding box annotations for a limited set of object categories, as well as image-caption pairs that cover a larger variety of objects at a significantly lower cost. We show that the proposed method can detect and localize objects for which no bounding box annotation is provided during training, at a significantly higher accuracy than zero-shot approaches. Meanwhile, objects with bounding box annotation can be detected almost as accurately as supervised methods, which is significantly better than weakly supervised baselines. Accordingly, we establish a new state of the art for scalable object

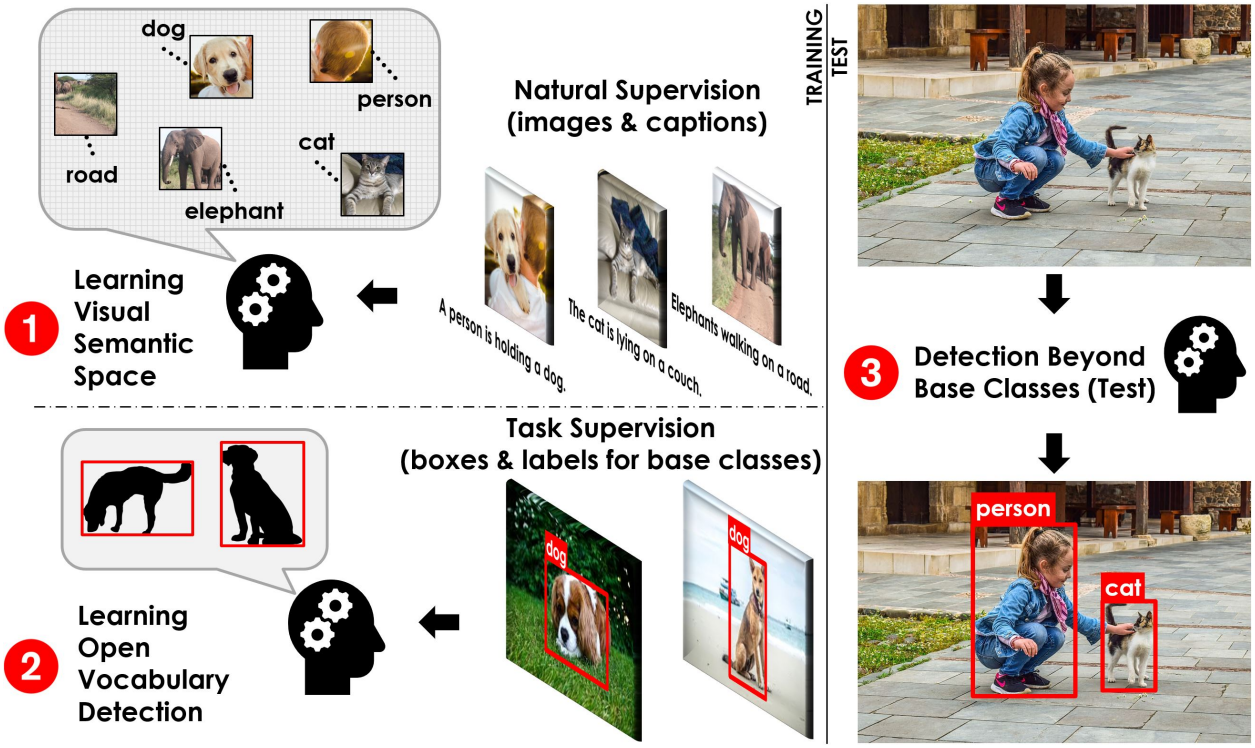


Figure 7.1: An overview of Open-Vocabulary Object Detection. We propose a two-stage training framework where we first (1) construct a visual-semantic space using low-cost image-caption pairs, and then (2) learn object detection using object annotations for a set of base classes. During test (3), the goal is to detect object categories beyond base classes, by exploiting the semantic space.

detection. This chapter including all images, figures, tables, equations, and text is based on a recently published collaborative work [159].

7.1 Introduction

Object detection is one of the most prominent applications of artificial intelligence, and one of the most successful tasks for deep neural networks. However, despite the tremendous progress in deep object detection, such as Faster R-CNN [3] and its impressive accuracy, training such models requires expensive and time-consuming human supervision. Particularly, one needs to manually annotate at least thousands of bounding boxes for each object category of interest. Although such efforts have been already made and there are valuable datasets publicly available, such as Open Images [160] and MSCOCO [161], these datasets cover a limited set of object categories (*e.g.* 600),

despite requiring extensive resources. Extending object detection from 600 to 60,000 categories requires 100 times more resources, which makes versatile object detection out of reach.

Nevertheless, humans learn to recognize and localize objects effortlessly through natural supervision, *i.e.*, exploring the visual world and listening to others describing situations. Their lifelong learning of visual patterns and associating them with spoken words results in a rich visual and semantic vocabulary that can be used not only for detecting objects, but for other tasks too, such as describing objects and reasoning about their attributes and affordances. Although drawing bounding boxes around objects is not a task that humans naturally learn, they can quickly learn it using few examples, and generalize it well to all types of objects, without needing examples for each object class.

In this chapter, we imitate this human ability, by designing a two-stage framework named Open-Vocabulary object Detection (OVD). We propose to first use a corpus of image-caption pairs to acquire an unbounded vocabulary of concepts, simulating how humans learn by natural supervision, and then use that knowledge to learn object detection (or any other downstream task) using annotation for only some object categories. This way, costly annotation is only needed for some categories, and the rest can be learned using captions, which are much easier to collect, and in many cases freely available on the web. Figure 7.1 illustrates the proposed OVD framework, which is novel and efficient, enables versatile real-world applications, and can be generalized to other computer vision tasks.

More specifically, we train a model that takes an image and detects any object within a given *target* vocabulary $\mathcal{V}_T$. To train such a model, we use an image-caption dataset covering a large variety of words denoted as $\mathcal{V}_C$ as well as a much smaller dataset with localized object annotations from a set of *base* classes $\mathcal{V}_B$. Note that in this task, target classes are not known during training, and can be any subset of the entire language vocabulary $\mathcal{V}_\Omega$. This is in contrast with most existing object detection settings including weakly supervised transfer learning methods, where $\mathcal{V}_T$ should be known beforehand. The most similar task to OVD is zero-shot object detection, which also generalizes to any given target set, but cannot utilize captions. Figure 7.2 illustrates an

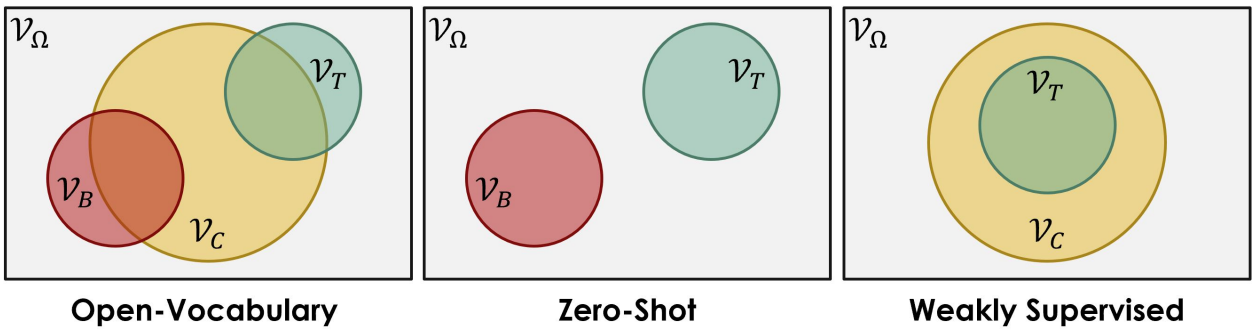


Figure 7.2: A comparison of our proposed OVD with existing ZSD and WSD paradigms. While zero-shot detection methods learn a limited set of base classes $\mathcal{V}_B$ and struggle to generalize to target classes $\mathcal{V}_T$, we acquire a much larger vocabulary $\mathcal{V}_C$ by learning from low-cost image-caption pairs. Although there are weakly supervised approaches that can learn from captions, they cannot use bounding box supervision from base classes, and need to know $\mathcal{V}_T$ before training. Hence, our OVD formulation is a generalization of ZSD and WSD, which can use both data sources to reach an outstanding performance on target classes not known in advance.

intuitive abstraction of our proposed task compared to zero-shot and weakly supervised detection. Despite close connections to those well-known ideas, OVD is novel and uniquely positioned in the literature, as we elaborate in Section 7.2.

To address the task of OVD, we propose a novel method based on Faster R-CNN [3], which is first pretrained on an image-caption dataset, and then fine-tuned on a bounding box dataset, in a particular way that maintains the rich vocabulary learned during pretraining, enabling generalization to object categories without annotation. Through extensive experiments, we evaluate our method, Open Vocabulary R-CNN (OVR-CNN), and show that it achieves significantly higher performance than the state of the art in zero-shot learning (27% mAP compared to 10%). We also show that it outperforms weakly supervised object detectors by a significant margin in generalized zero-shot settings (40% mAP compared to 26%). We provide comprehensive open-source code to reproduce results.[1]

7.2 Related work

Zero-shot object detection (ZSD) aims to generalize from annotated (seen) object classes to other (unseen) categories. The key idea is to use zero-shot learning techniques (*e.g*. word embedding projection [162]) to learn object proposal classification. Bansal *et al*. [55] argued the main challenge in ZSD is modeling the background class, which is hard to separate from unseen classes. They defined background as a mixture model, which was later improved by the introduction of polarity loss [163]. On the other hand, Zhu *et al*. [164, 165] argued the key to ZSD is to improve the generalization ability of object proposal models. They employed a generative model to hallucinate unseen classes and augment seen examples when training the proposal model. Nevertheless, ZSD methods are still far from practical performance, due to their unnecessarily harsh constraint, *i.e*., not having any example of unseen objects, and having to guess how they look like solely based on their word embeddings [55, 163, 165] or textual descriptions [166]. This has motivated recent papers to simplify the task by making unrealistic assumptions, such as the availability of test data during training [167], or the availability of unseen class annotations to filter images with unseen object instances [168]. Considering datasets with natural, weak supervision are abundant and cheap, we propose an alternative, more realistic problem: Besides annotated data for "seen" classes, we assume an image-caption dataset is available that covers a larger variety of objects with an open vocabulary. This allows us to achieve 27% mAP on unseen classes, compared to the 10% state of the art, without much extra annotation effort. To this end, we address the open problem of knowledge transfer from image-caption pretraining to object detection.

Weakly supervised object detection (WSD) is the most widely used approach to train object detectors without bounding box annotations, by using image-level labels instead. The main challenge of WSD is localization, as each label may refer to any object in the image. This problem is typically addressed using multiple instance learning, which is a well-studied topic [64, 169, 170]. Although image-level labels are easier to collect than bounding boxes, they still require manual

[1] `https://github.com/alirezazareian/ovr-cnn`

effort, and they are typically limited to a predefined taxonomy. In contrast, we use captions, which are more natural to annotate and often freely available on the web, while also featuring a rich, open vocabulary of concepts. Learning object detection from captions has been studied at a limited scale. Cap2Det [171] parses captions into multi-label classification targets, which can be used to train a WSD model. However, that requires image-level labels to train the caption parser, and is limited to a closed vocabulary. Amrani *et al*. [172] train a WSD model based on the presence of a predefined set of words in captions, which is similarly closed-vocabulary, and discards the rich semantic content of captions, which we exploit through transformers. In contrast, Sun *et al*. [173] and Ye *et al*. [174] aim to discover an open set of object classes from image-caption corpora, and learn detectors for each discovered class. A key limitation of all such WSD methods is their inferior object localization accuracy. In contrast, we disentangle object recognition and localization into two independent problems. We learn recognition using open-vocabulary captions, while learning localization using a fully annotated dataset from a small subset of classes.

Object detection using mixed supervision has been studied in order to exploit both weak and full supervision. However, most existing methods need bounding box annotations for all classes, and use weak supervision only as auxiliary data [175, 176, 177]. More similar to our work are those which transfer a detector trained on supervised base classes to weakly supervised target classes [178, 179, 180]. These methods usually lose performance on base classes as we show in Section 7.4. In contrast, we treat this problem as an opposite knowledge transfer process: Instead of training on base classes first, and transferring to target classes using weakly supervised learning, we first use captions to learn an open-vocabulary semantic space that includes target classes, and transfer that to the task of object detection via supervised learning. Another limitation of all weakly supervised and mixed-supervision methods is that they require image-level annotations within a predefined taxonomy, and they only learn those predefined classes. In contrast, we use captions which are open-vocabulary and also more prevalent on the web, and we learn to generalize to any set of target classes on demand, without having to know them beforehand. VirTex [181] is the only method that uses captions as well as object annotations to train a detector, but it needs annotation

for all object classes while we can generalize from a subset of annotated categories.

Visual grounding of referring expressions can be seen as an open-vocabulary object localization problem: Given an image and a noun phrase that refers to an object, usually within the context of a full caption, the goal is to localize the referred object in the image using a bounding box. We are inspired by the rich literature of weakly supervised visual grounding methods [142, 143, 144, 145] to design our image-caption pretraining technique. More specifically, we learn to map caption words to image regions, by learning a visual-semantic common space. However, such a mapping alone cannot be used to detect objects during inference when no caption is provided. Therefore, we propose to transfer visual grounding knowledge to the task of object detection through another phase of training.

Vision-language transformers Our framework of pretraining with image-captions and transferring the learned knowledge to the downstream task is inspired by the recent success of multimodal transformers [147, 182, 183, 184] following ViLBERT [1]. These methods train transformers in a self-supervised manner to take image-caption pairs as input and extract versatile features that can be fine-tuned on various downstream vision-language tasks, resulting in tremendous improvements. However, they have not been applied to object detection yet, since they need both image and caption as input, and also because they rely on a pretrained object detector to articulate the image before feeding into transformers. Recently, PixelBERT [185] removed the latter requirement by applying transformers directly on the feature map. We use the idea of PixelBERT as a building block of our image-caption pretraining. Additionally, we propose a novel technique to transfer the pretrained model to the task of open-vocabulary object detection.

7.3 Method

Figure 7.3 illustrates the architecture of our proposed method, which is based on a Faster R-CNN [3] trained in a zero-shot manner. More specifically, it is trained on a set of *base* classes $\mathcal{V}_B$, and tested on another set of *target* classes $\mathcal{V}_T$. To this end, pretrained word embeddings (*e.g.*

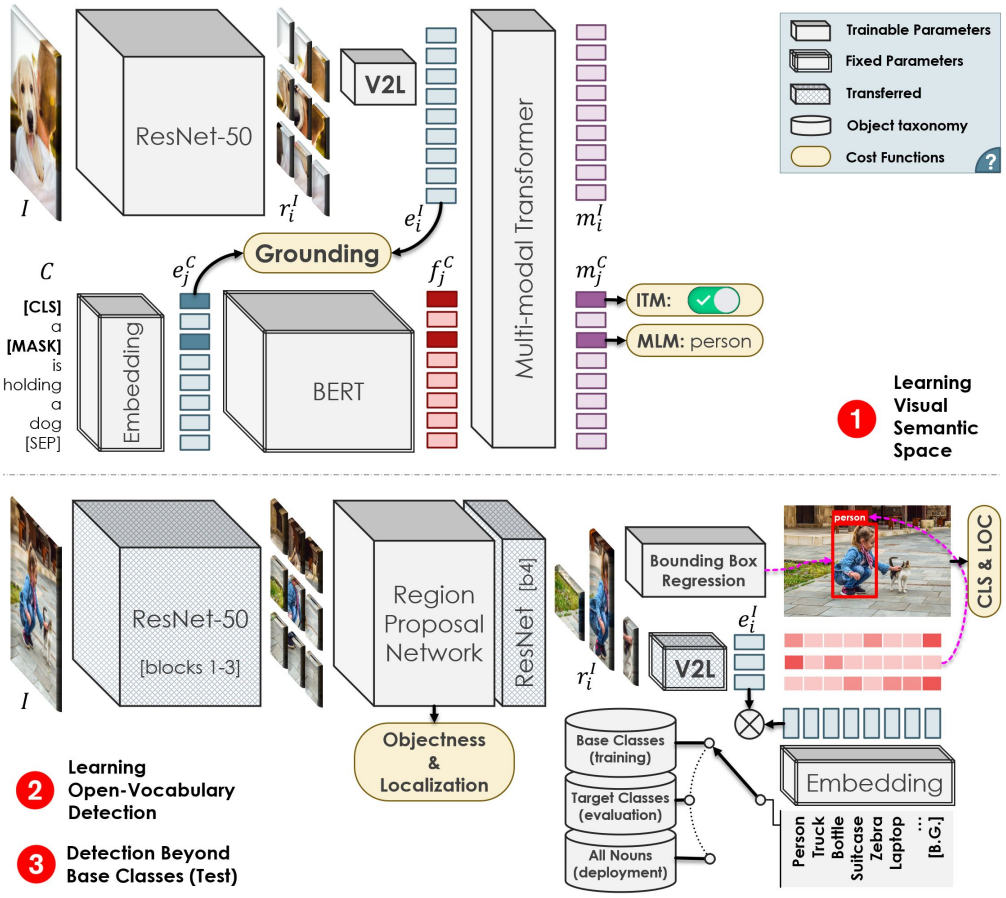


Figure 7.3: The architecture of our OVR-CNN during pretraining (top) and downstream training (bottom). We first train the ResNet and the V2L layer on image-caption pairs via grounding, masked language modeling (MLM) and image-text matching (ITM). Then we use the trained ResNet and V2L to initialize a Faster R-CNN in order to learn open-vocabulary object detection.

GloVE [74]) are often used instead of conventional, trainable classifiers, so that target class embeddings can replace base class embeddings during testing, without changing the model's output semantic space. Nevertheless, this practice often leads to severe overfitting due to the small sample of base classes, to the point where the state-of-the-art mAP on target classes is 9 times lower than base classes [163].

To alleviate this problem, our key idea is to pretrain the visual backbone on a larger vocabulary $\mathcal{V}_C$ to learn a more complete semantic space rather than a small number of base classes. Since captions are naturally written without much constraint on the vocabulary, they are a perfect source for learning a rich and complete visual-semantic space. We name this framework Open Vocabulary

Object Detection (OVD), as there are no explicit limits on the vocabulary of objects that can be learned through captions. In practice, our vocabulary is not literally "open", as it is limited to pretrained word embeddings. However, word embeddings are typically trained on very large text corpora such as Wikipedia that cover nearly every word [74, 97].

In the rest of this section, we elaborate how we pretrain our Open Vocabulary faster R-CNN (OVR-CNN) on image-caption pairs, and how we transfer the pretraining knowledge to the downstream task. In Section 7.4, we demonstrate that our method closes the base-target performance gap from a ratio of 9 to 2.

7.3.1 Learning a visual-semantic space

Object detectors typically use a CNN backbone that is often pretrained for ImageNet classification [186, 3]. Pretraining results in a backbone that can extract features optimized for object recognition, which is then used to train a new classification head for a fixed set of annotated classes. This is problematic in zero-shot settings, as a classifier trained on base classes cannot recognize target classes. Therefore, zero-shot methods learn a linear projection from visual features to pretrained base class embeddings by replacing classifier weights with a fixed embeddings matrix [162]. This way, the network is expected to generalize to target classes by assuming the continuity of the embedding space. Nevertheless, this approach is prone to overfitting, as projecting to a small number of the embedding space (base class embeddings) is an under-determined problem [55].

To prevent overfitting, we propose to learn the aforementioned Vision to Language (V2L) projection layer along with the CNN backbone during pretraining, where the data is not limited to a small set of base classes. To this end, we use an image-caption dataset, since captions contain a rich vocabulary and semantic structure that can be used to learn the meaning of words, including object names. To effectively learn from the rich supervision that captions provide, we exploit recent advances in visual grounding and vision-language transformers. We use a main (grounding) task as well as a set of auxiliary self-supervision tasks to learn a robust CNN backbone and V2L layer. In the next subsection, we elaborate how we transfer the pretrained modules to learn

open-vocabulary object detection.

Our pretraining architecture resembles PixelBERT [185]: it takes image-caption pairs as input, feeds the image into a visual backbone and the caption into a language backbone, which results in a set of token embeddings for the image and caption, and then feeds those token embeddings into a multimodal transformer to extract multimodal embeddings. Our visual backbone is a ResNet-50 [187], which takes a $w \times h$ image I as input and extracts a grid of $w/32 \times h/32$ regions, where each region i is represented by a d_v-dimensional feature vector, r_i^I. For the language backbone, we use a pretrained BERT [97], which takes a tokenized caption C as input, extracts a d_l-dimensional word embedding e_j^C for each token j, augments that with position embeddings, and applies several layers of multi-head self-attention to extract d_l-dimensional contextualized token embeddings f_j^C.

Furthermore, we devise a linear V2L layer that maps each visual region representation r_i^I into the language embedding space e_i^I. The final embeddings of image regions $\{e_i^I\}$ and caption tokens $\{f_j^C\}$ are then aggregated and fed into a multimodal transformer, which is similar to BERT in architecture, but performs attention not only within each modality but also across the two modelities. The output of the multimodal transformer is $\{m_i^I\}$ and $\{m_j^C\}$ for the regions and words respectively, which can be used for various pretraining tasks, as we discuss later in this section.

Once we extract the aforementioned stages of unimodal and multimodal embeddings from a batch of image-caption pairs, we define a main objective function as well as various auxiliary objectives to ensure an effective training for the ResNet parameters, as well as the V2L layer. Our main objective is visual grounding, *i.e.*, word embeddings from each caption e_j^C should be close to their corresponding image regions e_i^I. Since the correspondence of words and regions is not given, we employ a weakly supervised grounding technique to learn it, as illustrated in Figure 7.4. Specifically, we define a global grounding score for each image-caption pair, that is a weighted average of local grounding scores for word-region pairs:

$$\langle I, C \rangle_G = \frac{1}{n_C} \sum_{j=1}^{n_C} \sum_{i=1}^{n_I} a_{i,j} \langle e_i^I, e_j^C \rangle_L, \tag{7.1}$$

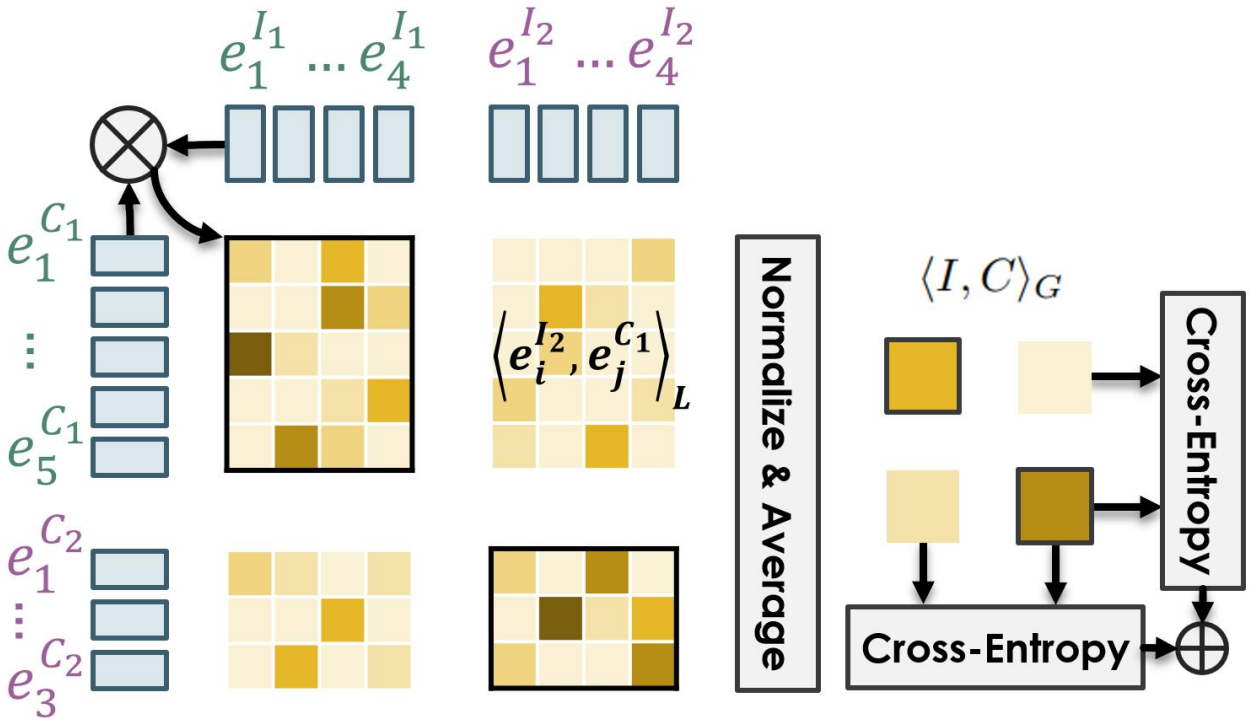


Figure 7.4: An illustration of our image-caption grounding method.

where $\langle ., . \rangle_L$ is the dot product of two vectors, n_I and n_C are the number of image and caption tokens, and

$$a_{i,j} = \frac{\exp\langle e_i^I, e_j^C \rangle_L}{\sum_{i'=1}^{n_I} \exp\langle e_{i'}^I, e_j^C \rangle_L}. \tag{7.2}$$

The global grounding score for a matching image-caption pair should be maximized, while it should be minimized for a non-matching pair. Hence, we use other images in the batch as negative examples for each caption, and use other captions in the batch as negative examples for each image. Accordingly, we define two grounding objective functions:

$$\mathcal{L}_G(I) = -\log \frac{\exp\langle I, C \rangle_G}{\sum_{C' \in \mathcal{B}_C} \exp\langle I, C' \rangle_G}, \tag{7.3}$$

and

$$\mathcal{L}_G(C) = -\log \frac{\exp\langle I, C \rangle_G}{\sum_{I' \in \mathcal{B}_I} \exp\langle I', C \rangle_G}, \tag{7.4}$$

where $\mathcal{B}_I$ and $\mathcal{B}_C$ are the image and caption batch. We validated the described formulation by

completing extensive experimentation with various other alternatives, such as other similarity metrics (*e.g.* cosine instead of dot product), other loss functions (*e.g.* triplet loss instead of negative log likelihood) and other word-to-region alignment mechanisms (*e.g.* hard alignment instead of softmax).

Optimizing the grounding objectives results in a learned visual backbone and V2L layer that can map regions in the image into words that best describe them, without limiting to a closed vocabulary. However, since we induce a weak, indirect supervision, a local optima might be achieved where the model only learns the minimum concepts necessary to choose the right image/caption. To more directly learn each word, we employ masked language modeling following PixelBERT [185]. Specifically, we randomly replace some words j in each caption C with a [MASK] token, and try to use the multimodal embedding of the masked token m_j^C to guess the word that was masked. To this end, the visual backbone and the V2L layer should learn to extract all objects that might be described in captions, and the multimodal transformer should learn to use those along with the language understanding ability of BERT to determine what word completes the caption best.

Accordingly, we apply a fully connected layer on m_j^C, compare its output to all word embeddings using dot product, and apply softmax to compute a probability score for each word. We define masked language modeling $\mathcal{L}_{MLM}$ as a cross-entropy loss comparing the predicted distribution with the actual word that was masked. PixelBERT also employs an image-text matching loss $\mathcal{L}_{ITM}$, but does not use masked visual modeling that is common in vision-language transformers [1]. We follow their choices for our auxiliary objectives, although other combinations are possible. We train the visual backbone, V2L layer, and the multimedia transformer jointly by minimizing the total loss for each image-caption pair:

$$\mathcal{L}(I, C) = \mathcal{L}_G(I) + \mathcal{L}_G(C) + \mathcal{L}_{MLM} + \mathcal{L}_{ITM}. \tag{7.5}$$

Note that our language backbone (BERT) and its word embeddings are fixed in our experiments.

7.3.2 Learning open-vocabulary detection

Once the ResNet visual backbone and V2L layer are trained, we transfer them to the task of object detection, by initializing a Faster R-CNN. Following [3], we use the stem and the first 3 blocks of our pretrained ResNet to extract a feature map from a given image. Next, a region proposal network slides anchor boxes on the feature map to predict objectness scores and bounding box coordinates, followed by non-max suppression and region-of-interest pooling to get a feature map for each potential object. Finally, following [3], the 4th block of our pretrained ResNet is applied on each proposal followed by pooling to get a final feature vector r_i^I for each proposal box, which is typically fed into a classifier in supervised settings.

Nevertheless, in our zero-shot setting, a linear layer is applied on the visual features r_i^I to map each proposal onto a word embedding space e_i^I, so they can be compared to base or target class embeddings in the training or testing phase respectively. In all ZSD methods, the aforementioned linear layer is trained from scratch on base classes, which struggles to generalize. In contrast, we have already trained the V2L layer in the pretraining phase, on a much broader semantic space. The main difference of this phase with pretraining is that instead of the grid-structured feature map, r_i^I represents a bounding box of arbitrary shape. However, due to the linear characteristics of RoI-Align [188], r_i^I is on the same space as in pretraining, with minimal domain shift that can be eliminated by fine-tuning the ResNet backbone.

During training, we compare e_i^I to each base class k to compute classification scores:

$$p(i \text{ classified as } k) = \frac{\exp\langle e_i^I, e_k^{\mathcal{V}} \rangle}{1 + \sum_{k' \in \mathcal{V}_B} \exp\langle e_i^I, e_{k'}^{\mathcal{V}} \rangle}, \tag{7.6}$$

where $e_k^{\mathcal{V}}$ is the pretrained embedding of word k, $\mathcal{V}_B$ is the set of base classes, and $\langle ., . \rangle$ denotes dot product. The addition of 1 in the denominator is because we set the background class to have a fixed, all-zero embedding, which makes any dot product zero, and is exponentiated to 1. We found that a fixed all-zero background embedding performs better than a trainable one as it does not push non-foreground bounding boxes, which may contain target classes, to an arbitrary region of the

embedding space.

Except for the aforementioned changes in the classification head, the rest of our network exactly follows Faster R-CNN, and is trained in the exact same way with the same objective functions. Empirically, we found that multiplying a ratio α to the classification loss of background proposals (*i.e.*, proposal boxes that are not matched with any ground truth bounding box) can improve the performance on target classes significantly, while slightly lowering base class performance. Hence, we use cross-validation to find the best α for each model. The ResNet parameters are finetuned, while the region proposal network and the regression head are trained from scratch. The classifier head is fully fixed, as it consists of a pretrained V2L layer and word embeddings, which are especially prone to overfitting. During testing, we use the model just like a Faster R-CNN, except we can replace word embeddings in Eq. (7.6) with any set of target classes $\mathcal{V}_T$. While we evaluate on a fixed, annotated target set, the model is not particularly tuned for those classes, and hence can be deployed on the entire vocabulary $\mathcal{V}_\Omega$.

7.4 Experiments

In this section, we demonstrate our method's ability to detect objects of the target classes accurately, while not losing its accuracy on the base classes compared to supervised approaches. Particularly, we show significant quantitative improvements compared to zero-shot and weakly supervised object detection methods, followed by a comprehensive analysis including ablation and visualization.

7.4.1 Data and metrics

We base our experiments on the challenging and widely used COCO Objects dataset [161]. We use their 2017 training and validation split for training and evaluation respectively. To select base and target classes, we adopt the split proposed by [55] and used by all other ZSD methods. Their splits includes 48 base classes and 17 target classes, which are both subsets of COCO object classes. We remove any bounding box that is not labeled with a base class from training data, and

remove images that are left with no bounding boxes. This leaves us with 107,761 training images that contain 665,387 instances of base classes, and 4,836 test images that contain 28,538 instances of base classes and 4,614 instances of target classes.

Unless otherwise mentioned, for pretraining we use COCO Captions [189], which is based on the same images and same train/test split as COCO Objects. This dataset is preferred due to the matching domain with the downstream task. However, to study more realistic settings, we also report results by pretraining on Conceptual Captions (CC) [190], which was automatically collected from the web. CC is larger with 2,749,293 training image-caption pairs, compared to COCO with 118,287 images and 5x captions.

Following most ZSD and WSD methods, we evaluate using mean Average Precision (mAP) at an intersection over union of 0.5. However, we report more comprehensive results at various evaluation settings. We evaluate our mAP on base classes by directly applying the model on COCO validation images and using base class annotations to evaluate. Then we replace the classifier head with target class embeddings and apply on COCO validation images, but this time compare with target class annotations. These result in base and target mAP, which resemble supervised and zero-shot settings respectively. We also replace the classifier head with the union of base and target class embeddings, to mimic generalized zero-shot settings [163]. In that case, we report total mAP, as well as separately computing the mean of AP over base and target classes.

7.4.2 Implementation details

We used the `maskrcnn-benchmark` code base [191], and particularly the `R_50_C4` configuration to implement our system. We also used a pretrained and frozen `BERT-Base` [192] as our language backbone. For the multimodal transformer, we use the same architecture as `BERT-Base`, except we use only 6 layers and 8 attention heads at each layer, and we train it from scratch. Our base learning rate for pretraining is 0.01 which drops to 0.001 and 0.0001 after sufficient training. We use a batch size of 64 and train on 8 V-100 GPUs which takes about 10 hours. We use spatial dropout following [185] to subsample visual regions during pretraining. For masked language

modeling, we mask each word with the likelihood of 0.135. We use gradient clipping at 5.0 for pretraining.

During downstream training, we use the BERT embeddings (*i.e.*, pretrained input embeddings, not the output of BERT transformers) of the base classes to initialize and fix the classifier weights. We found the best background weight is $\alpha = 0.2$ for most experiments, except the ablations without a fixed, pretrained V2L layer, where $\alpha = 0.0$ works best. We only fine-tune the third and forth block of ResNet, and keep the stem and first two blocks fixed. We train using a learning rate of 0.005 and drop to 0.0005 and 0.00005 when appropriate. We train with a batch size of 8 on 8 V-100 GPUs which takes about 18 hours to converge.

7.4.3 Baselines

We mainly compare to zero-shot detection methods, as ZSD is the closest area to our work. Particularly, we compare to SB [55], which is the first and simplest ZSD method, projecting CNN features of EdgeBox proposals [193] to word embeddings. Then we compare to LAB [55], which attempts to better model the background class using a mixture model. We also compare to DSES [55], which uses additional classes from Visual Genome [54] to augment base classes. Then we compare to PL [163], which proposes polarity loss to address the object-background imbalance, and to DELO [165], which employs a generative approach to prepare the model for certain target classes through feature hallucination. Note that DELO needs to know target classes beforehand, which makes it not truly open-vocabulary.

It is important to note that our approach utilizes extra data (COCO Captions or Conceptual Captions) that is not available to ZSD baselines. Although there is no method that works in the same setting as ours, we adopt weakly supervised detection (WSD) to our setting, by converting captions into image-level labels using exact matching or a classifier [171]. We compare to the well-known WSDDN [64], as well as Cap2Det [171] which better utilizes captions. However, WSD methods cannot utilize bounding box supervision for base classes. To have a fair comparison with stronger baselines, we also compare to transfer learning methods that utilize a mixture of weak

Table 7.1: Results on the MSCOCO dataset. Numbers are mAP (%). *For some baselines, target classes are known during training.

Method	Task	Base (48)	Target (17)	Generalized (48+17) Base	Target	All
FR-CNN [3]	FSD	54.5	-	-	-	-
WSDDN [64]*	WSD	-	-	19.6	19.7	19.6
Cap2Det [171]*		-	-	20.1	20.3	20.1
LSDA [178]*	MSD	-	-	29.3	17.7	27.2
LSDA+[179]*		-	-	28.5	21.9	26.7
MIL+RPN[180]*		-	-	27.8	22.6	26.4
SB [55]	ZSD	29.7	0.70	29.2	0.31	24.9
LAB [55]		21.1	0.27	20.8	0.22	18.0
DSES [55]		27.2	0.54	26.7	0.27	22.1
DELO [165]*		14.0	7.60	13.8	3.41	13.0
PL [163]		36.8	10.0	35.9	4.12	27.9
OVR-CNN	**OVD**	**46.8**	**27.5**	**46.0**	**22.8**	**39.9**

and full supervision (denoted as MSD). Particularly, we compare to LSDA [178], which learns a transformation from classifier weights into detector weights, its extension [179] to utilize semantic class relationships (LSDA+), and a more recent work [180] which uses multiple-instance learning on a region proposal network that is pretrained on base classes (MIL+RPN).

Note that since WSD and MSD methods require image-level labels, target classes should be known in advance during pretraining, and the models are particularly adapted to those classes. In contrast, our method and most ZSD methods have no access to such information, and can be applied to any novel class without retraining.

7.4.4 Results

Table 7.1 demonstrates our main results compared to the baselines. Particularly, we observe a significant improvement on target class performance and generalized target performance compared to all ZSD baselines. This is mainly due to our ability to utilize additional, low-cost training data. We also outperform WSD and MSD baselines on target classes, despite their unfair access to information about target classes during training, and we significantly outperform them on base

classes and therefore overall, due to our effective exploitation of bounding box supervision for base classes. Note that WSD and MSD models cannot be evaluated on base-only or target-only classes since they have a fixed classifier trained on all 65 classes. Moreover, we have a FSD (fully supervised detection) baseline to measure the performance drop on base classes.

Furthermore, we present ablation experiments in Table 7.2 to show the effectiveness of each design choice. Particularly, we observe that without pretraining our model on image-caption datasets, the model performs poorly. This confirms the remarkable efficacy of multimodal pretraining for open-vocabulary generalization. We also observe that grounding is the main component of pretraining, which has a much larger effect than the auxiliary objectives that are optimized through the multimedia transformer module. Moreover, we show that transferring ResNet weights alone (from pretraining to downstream task) is not enough for effective knowledge transfer, and we must transfer the V2L layer as well. Additionally, if the V2L layer is not frozen during downstream training, it loses its ability to generalize to target classes, in order to slightly improve on base classes.

We also try initializing the model randomly during pretraining instead of using widely used Imagenet weights, and despite the performance drop, we still perform better than most ZSD baselines that use Imagenet. We also observe that if we use the automatically collected Conceptual Captions instead of the carefully annotated COCO Captions, the performance drops, but still outperforms all ZSD baselines significantly, proving that even low-quality, cheap data can be utilized by OVR-CNN to achieve better performance.

7.4.5 Visualization

To gain deeper insight about what OVR-CNN learns, we depict the visual-semantic embedding space that is learned by our model in Figure 7.5. More specifically, we apply our trained model (after downstream training) on all COCO validation images, get the embeddings of all output bounding boxes after the V2L layer e_i^I, and reduce their dimentionality to 2 using t-SNE [194]. We color-code them based on their ground truth label and overlay class embeddings $e_k^{\mathcal{V}}$ on the same

Table 7.2: Ablation on MSCOCO dataset. Numbers are mAP (%).

Ablation	Base (48)	Target (17)	All (65)
Ours w/o pretraining	25.2	4.4	18.1
Ours w/o grounding	25.9	4.6	19.0
Ours w/o auxiliary objectives	45.6	26.0	38.8
Ours w/o transferring V2L	25.3	4.9	18.6
Ours w/o freezing V2L	47.0	23.4	39.3
Ours w/o Imagenet	18.4	9.13	14.3
Ours w/ Conceptual Captions	43.0	16.7	34.3
Ours	**46.8**	**27.5**	**39.9**

space. We only show target classes and their instances to reduce clutter. Ideally, instances of each target class should form distinct clusters, and each class embedding (prototype) should fall inside the cluster formed by its instances. This is particularly difficult to achieve for target classes due to the lack of direct supervision. We compare our method to a ZSD baseline that is identical to our model except without pretraining on image-caption pairs.

We observe that in the baseline, target classes form convoluted clusters and their prototypes are randomly distributed or collapsed. On the other hand, our full model creates well-defined clusters that contain their prototypes in most cases. This is consistent with our intuition and our quantitative results that suggest zero-shot learning is not sufficient for learning a smooth and generalizable mapping from visual features to semantic embeddings, and learning a larger vocabulary through multimodal data is crucial for a more coherent space and generalizing beyond base classes.

7.4.6 Discussion

Since one of the most critical issues of deep learning is bias, we analyze the effect of training data bias on our per-class performance. Since we have two training phases, the class frequency during pretraining and downstream training should be separately analyzed. Figure 7.6 shows our per-class performance (right), along with the frequency of bounding box instances during downstream training (left), and the frequency of words during pretraining (center).

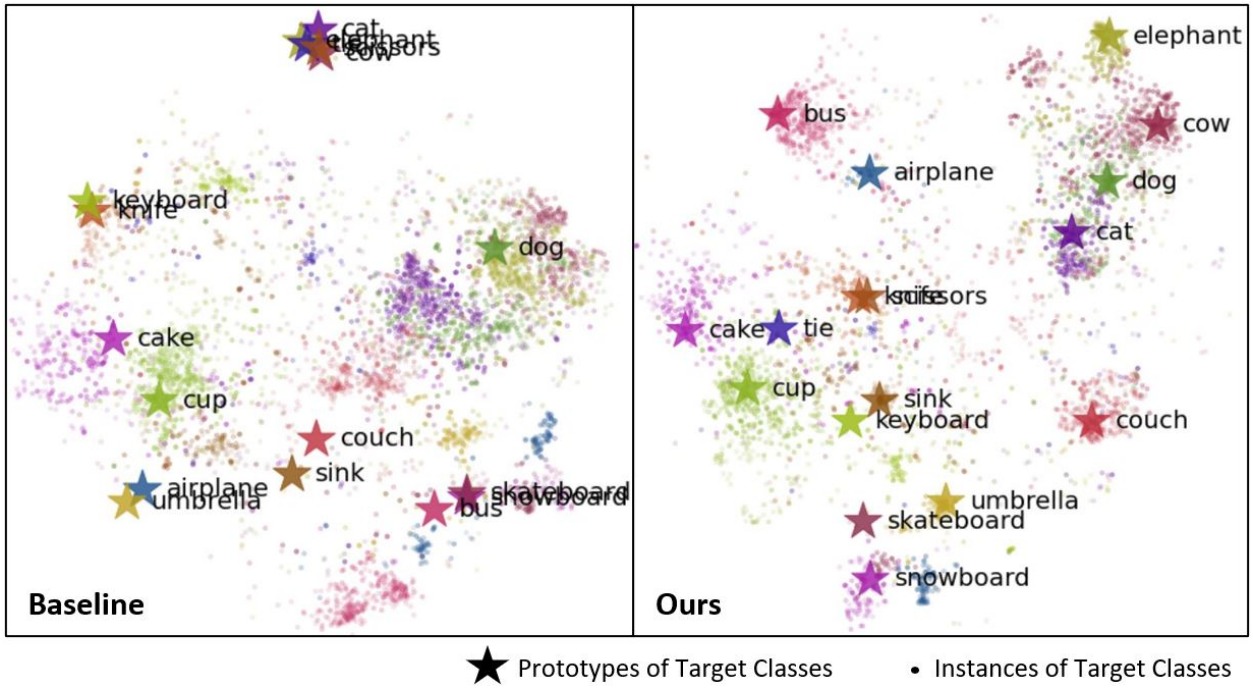


Figure 7.5: The embedding space learned by OVR-CNN (right) compared to a baseline without pretraining (left). Each color represents a target class, each dot represents the e_i^I embedding of a bounding box and each star represents a class prototype.

Our first observation is that our performance is not affected by the bias in downstream training data. As we move down the list, classes become exponentially less frequent, but the performance does not drop at all, except target (red) classes which have exactly zero examples during downstream training, and are inevitably less accurate. Our robustness to data bias is most likely due to the fact that we fix the classification head during downstream training, including both the V2L layer and the class embeddings. This is in contrast with conventional classifiers which fully adapt the classifier parameters, including an explicit *bias* term, to the biased training data.

Nevertheless, when we compare the performance to word frequency during pretraining, we do observe a correlation between the least frequent words and the lest accurate classes. This correlation is not very strong, but it motivates our future work on bias mitigation mechanisms that can be used in naturally supervised (image-caption) settings.

Furthermore, we observe that smaller objects such as `knife` and `tie` have lower performance, which is to some extent consistent with supervised object detection, but is fueled by the fact that

our grounding mechanism is weakly supervised, and is less likely to correctly align smaller objects to words, because they take a smaller portion of the feature map.

7.4.7 Qualitative results

To get a qualitative look at the performance, we deploy our model on the COCO validation set and visualize its detection outputs in Figure 7.7. We use the generalized version which selects the category of each object from the union of base and target classes. We emphasize target classes for better visibility, and analyse the quality of the predictions. Based on our observation, the main limitation of our method is localization accuracy for target classes. There are several cases of overly loose or overly tight bounding boxes, which is due to the fact that we have no ground truth bounding boxes for target classes. This motivates future work on class-agnostic boundary refinement.

7.5 Summary

We called attention to the new task of Open-Vocabulary Object Detection (OVD), as an attempt to disentangle object detection into recognition and localization, and learn them separately using two different sources of supervision that are perfect for each corresponding task. In OVD, recognition is learned from captions, which are general-purpose and open-vocabulary, while localization is learned from bounding box annotations, which are accurate and directly designed for the downstream task. We proposed OVR-CNN which pretrains a Faster R-CNN on an image-caption dataset and carefully transfers the open-vocabulary visual-semantic knowledge learned from captions to the downstream task of object detection. We demonstrated record performance compared to zero-shot and weakly supervised baselines, establishing a new state of the art for scalable object detection. Nevertheless, OVR-CNN is merely one possible implementation of our general idea, which can be extended to other downstream tasks too, enabling more human-like, open-vocabulary computer vision technology.

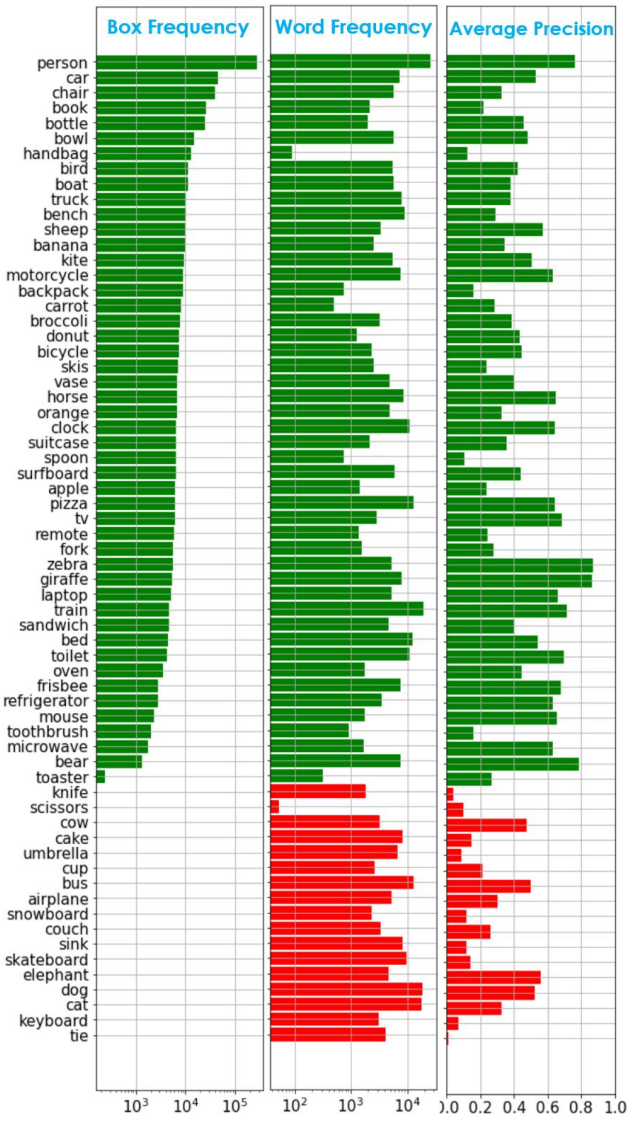


Figure 7.6: Performance for each class along with data frequency during pretraining and downstream training. Green and red show base and target classes respectively.

Figure 7.7: Qualitative results of our OVR-CNN model, detecting both base and target classes. Target classes are shown with larger font, thicker border, and uppercase.

Chapter 8: Conclusion and Open Problems

8.1 Summary of contributions

In this thesis, we developed new methods for creating deep learning models that can understand visual and multimedia content, by extracting structured symbolic representations. To this end, we addressed fundamental challenges such as weak supervision, limited vocabulary, and machine common sense, in order to develop more robust and scalable deep learning models. More specifically, we introduced the new tasks of VSP (Chapter 3) and its multimedia extension M^2E^2(Chapter 6), which aim to extract comprehensive semantic graphs from images and multimedia data respectively. Then we developed new methods such as VSPNet (Chapter 3) and WASE (Chapter 6), in order to learn to efficiently and effectively extract such structures from data. Moreover, we improved the accuracy of scene graph extraction methods by exploring two different ways to reinforce them with common sense knowledge, both using external knowledge graphs (Chapter 4) and by directly learning from data (Chapter 5). Finally, we proposed an extension to open-vocabulary settings, where naturally supervised, image-caption data is used to learn a broader range of concepts without considerable annotation cost, and that knowledge is transferred to down-stream scene understanding tasks such as object detection (Chapter 7).

Collectively, these findings bolster the functionality, robustness, and scalability of visual and multimedia understanding systems, and AI applications at large. We summarize the contributions and main published papers in the following:

- Introduced Visual Semantic Parsing (VSP), a new task to advance graph-based scene representations. Proposed VSPNet, a new method for extracting VSP graphs from images with subquadratic computational complexity. Created the first graph-based weakly supervised learning framework based on a novel graph alignment algorithm. This framework can ex-

tract more expressive representations from images that are beyond the capability of existing scene graph generation models, and do this with competitive accuracy while at least 5 times faster than the state of the art, and without requiring conventional but costly bounding box annotations [57].

- Introduced Graph Bridging Networks (GBNet), a new neural network architecture that takes an image and an external knowledge graph as input, and processes both inputs jointly in order to extract a scene graph that follows common sense knowledge. GBNet achieves an average of 25% relatively higher accuracy over the state of the art [77].

- Introduced Global Local Attention Transformers (GLAT), a new neural network architecture that takes a noisy scene graph as input and creates a more accurate scene graph that follows the commonsense knowledge it has learned from data. This simple process can considerably improve scene understanding accuracy without requiring any additional data or supervision [93].

- Introduced Multimedia Event Extraction (M^2E^2), a new task extending graph-based event understanding to multimedia data, including text and images. Proposed Weakly Aligned Structured Embeddings (WASE), a new network architecture for extracting semantic graphs that are agnostic to modality. Created a weakly supervised multi-task learning framework to learn M^2E^2without explicit multimedia data annotations. The proposed framework can extract more comprehensive information from multimedia documents compared to traditional single-modality methods, with a higher accuracy compared to unstructured baselines [112].

- Introduced Open-Vocabulary Object Detection (OVD), a new paradigm for scalable object detection using partial supervised data and large-scale naturally supervised image-caption pairs. Proposed Open-Vocabulary Faster R-CNN (OVR-CNN), which learns a visual semantic embedding space from image-caption pairs, and then uses that to learn object detection without annotation for all object classes. This improves the state-of-the-art mean average precision for zero-shot object detection from 10% to 28% with little additional cost [159].

8.2 Open problems and future work

We focused on a few critical challenges in scene understanding and multimedia event extraction, and conducted research that resulted in more accurate, more robust, more scalable, and more comprehensive scene understanding technology. Nevertheless, there are still many directions that demand further research. In order to use graph-based scene understanding in critical applications such as autonomous driving and healthcare, we need to ensure this technology is mature enough and robust enough to be used with a controlled amount of error. For more versatile applications such as robotics and augmented reality, we need models with a broader range of concepts and consistent performance across long-tailed distributions. For applications that require more in-depth reasoning such as multimedia dialog and question answering, we need more comprehensive representations beyond predicates and their argument roles. Moreover, more research is required for applying this technology for multimedia domains besides news, such as healthcare, and to incorporate other forms of data, such as speech and video. In the following, we envision a few particularly interesting future research directions in more detail.

Extension to videos: Static images are merely snapshots of what happens in the real world, and convey a fraction of the information found in videos. Extending graph-based scene understanding technology to videos involves a new temporal dimension that is missing from images, which brings important but fascinating new challenges. One obvious solution is to extract scene graphs at every frame separately, but that fails to incorporate temporal information. Preliminary research has been done to incorporate motion for SGG [195, 196], but this is still an open problem. Moreover, extracting scene graphs from videos enables the study of how they change over time, to understand the internal structure of higher-level events [197], and to gain temporal commonsense in a data-driven fashion. Furthermore, videos usually include an audio modality, which often contain speech, which can be further incorporated both as a source of natural supervision [198], and as a multimedia input to jointly extract information, similar to M^2E^2. Finally, fully understanding videos requires a broader range of semantic interactions, such as predicate-predicate relationships

(e.g. causality) and entity-entity coreference relations, which needs more complex graphical structures and extraction mechanisms.

More comprehensive representations: Our VSP and M^2E^2formulations expand the representation power of graph-based visual understanding methods, as they can represent situations beyond the capability of alternatives, such as SGG counterparts. For instance, VSP can represent a flexible number of entities involved in a predicate with various semantic roles, while SGG can only represent exactly two entities taking exactly two roles in each predicate. Nevertheless, there are still a broad range of semantic meanings that cannot be represented using VSP and M^2E^2. Fortunately, the NLP community is years ahead of CV in semantic representation research, which has led to a variety of powerful graph-based schema [11], such as Abstract Meaning Representation [151]. Extracting AMR graphs from visual and multimedia content has not been studied, but can potentially enable significantly richer structures beyond entity-predicate relationships, such as adjectives, adverbs, coreference resolution, etc. On the other hand, NLP-driven representations may not be optimal for representing visual content, and further research is needed to design a schema for visual and multimedia semantic graphs.

Higher-level concepts: The foundations of computer vision were built from the lowest level of understanding, such as edge detection and local feature analysis, to higher-level concepts such as objects and actions. Nevertheless, from a linguistic point of view, objects and actions are still atomic building blocks of higher-level semantics, such as interactions and situations. Whereas a birthday cake is already hard to detect due to complex visual structure and variations, a birthday party is an even more complex concept, involving a variety of objects and activities, with a more flexible internal structure. Similarly, while hugging is a visually non-trivial interaction between two people due to visual variations in pose and viewpoint, bullying is a far more abstract and complicated interaction that cannot be described by a handful of patterns. CV research has made inspiring progress towards low-level concepts such as objects and actions, but is far from understanding complex situations. Preliminary work on complex events has been reported [199], and

more recently extended to complex interactions and relationships [200]. However, these directions have not seen wide attention from the community yet.

Datasets, ontologies, and scope: There are few datasets with semantic graph annotations for visual and multimedia content, compared to other computer vision tasks, and compared to semantic parsing datasets in NLP. This may partly be due to the more costly and complex annotation process, but we argue that part of the problem is the lack of a proper scope and ontology for defining such datasets. Visual Genome [75] is one of the only datasets available for training SGG models, with many fundamental problems that have not been resolved yet, such as the quality and completeness of annotation, the severe bias in predicate frequencies, and the lack of a structured ontology to model the similarities of predicate categories. Moreover, if we were to create a new dataset, it is highly non-trivial how to choose categories, define a schema, and set a limit on the scope and comprehensiveness of annotations. Nevertheless, creating more progressive datasets with graph-based annotations is essential for the advance of graph-based representation methods.

More scalable training: Although the focus of this thesis is learning graph-based representations for more powerful scene understanding, and not reducing supervision, we believe developing more powerful AI models is barely useful unless scalable training methods are developed in conjunction. Therefore, throughout this thesis, we have supplemented our methodology with training algorithms that need less supervision than conventional deep learning methods (Chapters 3, 6, and 7). Nevertheless, there is still a long way to autonomous models that can learn comprehensive visual understanding through natural supervision, which is an effortless function of the human brain. One possible direction is to extend our new open-vocabulary detection paradigm to other vision tasks, such as visual semantic parsing, which requires not only open-vocabulary objects, but also open-vocabulary predicates.

Application: Scene graphs have already been proven useful for many downstream tasks, such as image captioning [201], image retrieval [16], and visual question answering [202]. Nevertheless,

they have not been fully utilized in all tasks that require visual and multimedia understanding. For instance, the state of the art in visual question answering and visual commonsense reasoning still relies on unstructured large-scale transformers, which reduce images into simplistic bag-of-words representations, without fully understanding the content [1, 148]. Part of the reason for the slow progress in this direction is that accurate and practical graph-based scene understanding methods are still out of reach, which is a bottleneck for developing downstream use cases. Nevertheless, this limitation will likely be alleviated soon, considering the vast amount of effort concurrent to this thesis for improving graph-based scene understanding.

8.3 Broader impact and ethical considerations

Like any technology, AI can be utilized with ethical or unethical intentions, and may have potential benefits and harms [203, 204, 205]. We acknowledge that our developed technology can cause potential harms in various different aspects, and we discuss the most important considerations in this section. Nevertheless, it should be noted that any potential harm we discuss here broadly applies to AI and CV research, and our intention in fulfilling this thesis has only been to benefit humanity through the vast amount of peaceful AI applications. Moreover, it is important to distinguish ethical issues related to the intentional misuse of our technology, from potential issues that may occur unintentionally as a result of irresponsible or careless utilization.

The most obvious ethical concern is to deliberately use AI technology for developing weaponry, spyware, or other tools that harm humanity. Visual and multimedia understanding can facilitate the scalable inspection of people's private data without their consent, or to analyze video surveillance with hostile intentions. It can also be utilized to develop weaponized autonomous robots that can understand situations to navigate and identify targets. To prevent such applications, international regulations must be adhered and further developed, such as the GDPR[1] regulations on obtaining and utilizing data, and the transparent audit of usage scenarios when deploying AI solutions or systems in practice.

[1]The General Data Protection Regulation of the European Union `https://gdpr.eu/what-is-gdpr/`

Nevertheless, unintentional ethical issues are also very important, since they are less trivial and harder to prevent. It is extremely important to understand the limitations of our technology and AI in general, before deployment on critical applications. For instance, the accuracy of our models are still limited, and may be further reduced in the face of adversarial attacks [206]. Hence, any use case in critical applications such as autonomous driving and healthcare must be performed with extreme care and comprehensive testing standards, or by ensuring a manual verification process involving humans as final decision makers rather than fully autonomous deployment. Moreover, AI is known to be prone to data bias [207, 208]. Hence, any beneficial use of this technology even with the best intentions may inadvertently harm less represented populations. Our developed systems, including the publicly available code and pretrained models, are in no way protected from those issues, and hence must be responsibly used. For instance, while the overall accuracy numbers that are reported in each chapter may sound adequate of a certain application, there might be a considerable performance variance across object and predicate categories.

Furthermore, in some applications, it is not trivial to balance the potential benefits and harms of AI technology, and careful considerations are required to optimize the overall benefit. For instance surveillance has potential benefits, such as preventing terrorism via security cameras and the spread of violent content or misinformation through social media. Nevertheless, surveillance without adequate privacy, security, and fairness protection measures may also cause harms, such as limiting the freedom of speech, jeopardizing democracy by inducing unintended but large-scale bias, or even harming vulnerable communities. Moreover, despite the influential success of large-scale models trained on public data, data privacy may be unintentionally violated by collecting public data for training, without paying attention to applicable guidelines and regulations.

References

[1] J. Lu, D. Batra, D. Parikh, and S. Lee, "Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks," in *Advances in Neural Information Processing Systems*, 2019, pp. 13–23.

[2] I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio, *Deep learning*, 2. MIT press Cambridge, 2016, vol. 1.

[3] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in neural information processing systems*, 2015, pp. 91–99.

[4] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014.

[5] D. Hendrycks, K. Zhao, S. Basart, J. Steinhardt, and D. Song, "Natural adversarial examples," *arXiv preprint arXiv:1907.07174*, 2019.

[6] A. Nguyen, J. Yosinski, and J. Clune, "Deep neural networks are easily fooled: High confidence predictions for unrecognizable images," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 427–436.

[7] Y. Goyal, T. Khot, D. Summers-Stay, D. Batra, and D. Parikh, "Making the v in vqa matter: Elevating the role of image understanding in visual question answering," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 6904–6913.

[8] J. Johnson, B. Hariharan, L. van der Maaten, L. Fei-Fei, C Lawrence Zitnick, and R. Girshick, "Clevr: A diagnostic dataset for compositional language and elementary visual reasoning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 2901–2910.

[9] R. Zellers, Y. Bisk, A. Farhadi, and Y. Choi, "From recognition to cognition: Visual commonsense reasoning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 6720–6731.

[10] S. Lapuschkin, S. Wäldchen, A. Binder, G. Montavon, W. Samek, and K.-R. Müller, "Unmasking clever hans predictors and assessing what machines really learn," *Nature communications*, vol. 10, no. 1, p. 1096, 2019.

[11] O. Abend and A. Rappoport, "The state of the art in semantic representation," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, vol. 1, 2017, pp. 77–89.

[12] A. Sharma, N. H. Vo, S. Aditya, and C. Baral, "Towards addressing the winograd schema challenge—building and using a semantic parser and a knowledge hunting module," in *Twenty-Fourth International Joint Conference on Artificial Intelligence*, 2015.

[13] D. Teney, L. Liu, and A. van den Hengel, "Graph-structured representations for visual question answering," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 1–9.

[14] J. Shi, H. Zhang, and J. Li, "Explainable and explicit visual reasoning over scene graphs," *arXiv preprint arXiv:1812.01855*, 2018.

[15] T. Yao, Y. Pan, Y. Li, and T. Mei, "Exploring visual relationship for image captioning," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 684–699.

[16] J. Johnson, R. Krishna, M. Stark, L.-J. Li, D. Shamma, M. Bernstein, and L. Fei-Fei, "Image retrieval using scene graphs," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3668–3678.

[17] G. E. Hinton, J. L. McClelland, D. E. Rumelhart, *et al.*, *Distributed representations*. Carnegie-Mellon University Pittsburgh, PA, 1984.

[18] Z. Shou, J. Chan, A. Zareian, K. Miyazawa, and S.-F. Chang, "Cdc: Convolutional-deconvolutional networks for precise temporal action localization in untrimmed videos," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 5734–5743.

[19] M. Gardner, P. Dasigi, S. Iyer, A. Suhr, and L. Zettlemoyer, "Neural semantic parsing," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics: Tutorial Abstracts*, 2018, pp. 17–18.

[20] J. Flanigan, S. Thomson, J. Carbonell, C. Dyer, and N. A. Smith, "A discriminative graph-based parser for the abstract meaning representation," in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2014, pp. 1426–1436.

[21] M. Li, Y. Lin, J. Hoover, S. Whitehead, C. Voss, M. Dehghani, and H. Ji, "Multilingual entity, relation, event and human value extraction," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, 2019, pp. 110–115.

[22] D. Wadden, U. Wennberg, Y. Luan, and H. Hajishirzi, "Entity, relation, and event extraction with contextualized span representations," *arXiv preprint arXiv:1909.03546*, 2019.

[23] A. Fader, L. Zettlemoyer, and O. Etzioni, "Open question answering over curated and extracted knowledge bases," in *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, ACM, 2014, pp. 1156–1165.

[24] D. Khashabi, T. Khot, A. Sabharwal, and D. Roth, "Question answering as global reasoning over semantic abstractions," in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[25] L. Dietz, A. Kotov, and E. Meij, "Utilizing knowledge graphs for text-centric information retrieval," in *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, ACM, 2018, pp. 1387–1390.

[26] J. Yu, Y. Lu, Z. Qin, W. Zhang, Y. Liu, J. Tan, and L. Guo, "Modeling text with graph convolutional network for cross-modal information retrieval," in *Pacific Rim Conference on Multimedia*, Springer, 2018, pp. 223–234.

[27] C. Lu, R. Krishna, M. Bernstein, and L. Fei-Fei, "Visual relationship detection with language priors," in *European Conference on Computer Vision*, Springer, 2016, pp. 852–869.

[28] H. Zhang, Z. Kyaw, S.-F. Chang, and T.-S. Chua, "Visual translation embedding network for visual relation detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 5532–5540.

[29] H. Zhang, Z. Kyaw, J. Yu, and S.-F. Chang, "Ppr-fcn: Weakly supervised visual relation detection via parallel pairwise r-fcn," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 4233–4241.

[30] X. Liang, L. Lee, and E. P. Xing, "Deep variation-structured reinforcement learning for visual relationship and attribute detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 848–857.

[31] B. Dai, Y. Zhang, and D. Lin, "Detecting visual relationships with deep relational networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 3076–3086.

[32] B. A. Plummer, A. Mallya, C. M. Cervantes, J. Hockenmaier, and S. Lazebnik, "Phrase localization and visual relationship detection with comprehensive image-language cues," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 1928–1937.

[33] R. Yu, A. Li, V. I. Morariu, and L. S. Davis, "Visual relationship detection with internal and external linguistic knowledge distillation," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 1974–1982.

[34] B. Zhuang, L. Liu, C. Shen, and I. Reid, "Towards context-aware interaction recognition for visual relationship detection," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 589–598.

[35] S. Jae Hwang, S. N. Ravi, Z. Tao, H. J. Kim, M. D. Collins, and V. Singh, "Tensorize, factorize and regularize: Robust visual relationship learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 1014–1023.

[36] J. Peyre, J. Sivic, I. Laptev, and C. Schmid, "Weakly-supervised learning of visual relations," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 5179–5188.

[37] D. Xu, Y. Zhu, C. B. Choy, and L. Fei-Fei, "Scene graph generation by iterative message passing," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 5410–5419.

[38] G. Gkioxari, R. Girshick, P. Dollár, and K. He, "Detecting and recognizing human-object interactions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 8359–8367.

[39] Y.-W. Chao, Y. Liu, X. Liu, H. Zeng, and J. Deng, "Learning to detect human-object interactions," in *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, IEEE, 2018, pp. 381–389.

[40] K. Kato, Y. Li, and A. Gupta, "Compositional learning for human object interaction," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 234–251.

[41] S. Qi, W. Wang, B. Jia, J. Shen, and S.-C. Zhu, "Learning human-object interactions by graph parsing neural networks," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 401–417.

[42] M. Yatskar, L. Zettlemoyer, and A. Farhadi, "Situation recognition: Visual semantic role labeling for image understanding," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 5534–5542.

[43] M. Yatskar, V. Ordonez, L. Zettlemoyer, and A. Farhadi, "Commonly uncommon: Semantic sparsity in situation recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 7196–7205.

[44] A. Mallya and S. Lazebnik, "Recurrent models for situation recognition," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 455–463.

[45] R. Li, M. Tapaswi, R. Liao, J. Jia, R. Urtasun, and S. Fidler, "Situation recognition with graph neural networks," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 4173–4182.

[46] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu, "A comprehensive survey on graph neural networks," *arXiv preprint arXiv:1901.00596*, 2019.

[47] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1609.02907*, 2016.

[48] Y. Li, W. Ouyang, B. Zhou, K. Wang, and X. Wang, "Scene graph generation from objects, phrases and region captions," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 1261–1270.

[49] Y. Li, W. Ouyang, B. Zhou, J. Shi, C. Zhang, and X. Wang, "Factorizable net: An efficient subgraph-based framework for scene graph generation," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 335–351.

[50] J. Yang, J. Lu, S. Lee, D. Batra, and D. Parikh, "Graph r-cnn for scene graph generation," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 670–685.

[51] K. Marino, R. Salakhutdinov, and A. Gupta, "The more you know: Using knowledge graphs for image classification," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[52] X. Wang, Y. Ye, and A. Gupta, "Zero-shot recognition via semantic embeddings and knowledge graphs," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 6857–6866.

[53] C.-W. Lee, W. Fang, C.-K. Yeh, and Y.-C. Frank Wang, "Multi-label zero-shot learning with structured knowledge graphs," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 1576–1585.

[54] R. Krishna, Y. Zhu, O. Groth, J. Johnson, K. Hata, J. Kravitz, S. Chen, Y. Kalantidis, L.-J. Li, D. A. Shamma, *et al.*, "Visual genome: Connecting language and vision using crowdsourced dense image annotations," *International Journal of Computer Vision*, vol. 123, no. 1, pp. 32–73, 2017.

[55] A. Bansal, K. Sikka, G. Sharma, R. Chellappa, and A. Divakaran, "Zero-shot object detection," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 384–400.

[56] L. Shen, S. Yeung, J. Hoffman, G. Mori, and L. Fei-Fei, "Scaling human-object interaction recognition through zero-shot learning," in *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, IEEE, 2018, pp. 1568–1576.

[57] A. Zareian, S. Karaman, and S.-F. Chang, "Weakly supervised visual semantic parsing," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 3736–3745.

[58] R. Zellers, M. Yatskar, S. Thomson, and Y. Choi, "Neural motifs: Scene graph parsing with global context," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 5831–5840.

[59] S. Woo, D. Kim, D. Cho, and I. S. Kweon, "Linknet: Relational embedding for scene graph," in *Advances in Neural Information Processing Systems*, 2018, pp. 558–568.

[60] A. Newell and J. Deng, "Pixels to graphs by associative embedding," in *Advances in neural information processing systems*, 2017, pp. 2171–2180.

[61] Y. Liu, R. Wang, S. Shan, and X. Chen, "Structure inference net: Object detection using scene-level context and instance-level relationships," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 6985–6994.

[62] X. Chen, L.-J. Li, L. Fei-Fei, and A. Gupta, "Iterative visual reasoning beyond convolutions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7239–7248.

[63] Y. Yuan, X. Liang, X. Wang, D.-Y. Yeung, and A. Gupta, "Temporal dynamic graph lstm for action-driven video object detection," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 1801–1810.

[64] H. Bilen and A. Vedaldi, "Weakly supervised deep detection networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2846–2854.

[65] Z. Shou, H. Gao, L. Zhang, K. Miyazawa, and S.-F. Chang, "Autoloc: Weakly-supervised temporal action localization in untrimmed videos," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 154–171.

[66] T. G. Dietterich, R. H. Lathrop, and T. Lozano-Pérez, "Solving the multiple instance problem with axis-parallel rectangles," *Artificial intelligence*, vol. 89, no. 1-2, pp. 31–71, 1997.

[67] M. Palmer, D. Gildea, and N. Xue, "Semantic role labeling," *Synthesis Lectures on Human Language Technologies*, vol. 3, no. 1, pp. 1–103, 2010.

[68] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using rnn encoder-decoder for statistical machine translation," *arXiv preprint arXiv:1406.1078*, 2014.

[69] T. K. Moon, "The expectation-maximization algorithm," *IEEE Signal processing magazine*, vol. 13, no. 6, pp. 47–60, 1996.

[70] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[71] J. Munkres, "Algorithms for the assignment and transportation problems," *Journal of the society for industrial and applied mathematics*, vol. 5, no. 1, pp. 32–38, 1957.

[72] A. Kuznetsova, H. Rom, N. Alldrin, J. Uijlings, I. Krasin, J. Pont-Tuset, S. Kamali, S. Popov, M. Malloci, T. Duerig, *et al.*, "The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale," *arXiv preprint arXiv:1811.00982*, 2018.

[73] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1026–1034.

[74] J. Pennington, R. Socher, and C. Manning, "Glove: Global vectors for word representation," in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 2014, pp. 1532–1543.

[75] S. Gupta and J. Malik, "Visual semantic role labeling," *arXiv preprint arXiv:1505.04474*, 2015.

[76] B. Wan, D. Zhou, Y. Liu, R. Li, and X. He, "Pose-aware multi-level feature network for human object interaction detection," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 9469–9478.

[77] A. Zareian, S. Karaman, and S.-F. Chang, "Bridging knowledge graphs to generate scene graphs," in *Computer Vision – ECCV 2020*, Springer International Publishing, 2020, pp. 606–623, ISBN: 978-3-030-58592-1.

[78] T. Chen, W. Yu, R. Chen, and L. Lin, "Knowledge-embedded routing network for scene graph generation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 6163–6171.

[79] J. Gu, H. Zhao, Z. Lin, S. Li, J. Cai, and M. Ling, "Scene graph generation with external knowledge and image reconstruction," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 1969–1978.

[80] H. Liu and P. Singh, "Conceptnet—a practical commonsense reasoning tool-kit," *BT technology journal*, vol. 22, no. 4, pp. 211–226, 2004.

[81] G. A. Miller, "Wordnet: A lexical database for english," *Communications of the ACM*, vol. 38, no. 11, pp. 39–41, 1995.

[82] Y. Zhao and S.-C. Zhu, "Image parsing with stochastic scene grammar," in *Advances in Neural Information Processing Systems*, 2011, pp. 73–81.

[83] M. Pei, Y. Jia, and S.-C. Zhu, "Parsing video events with goal inference and intent prediction," in *2011 International Conference on Computer Vision*, IEEE, 2011, pp. 487–494.

[84] K. Tu, M. Meng, M. W. Lee, T. E. Choe, and S.-C. Zhu, "Joint video and text parsing for understanding events and answering queries," *IEEE MultiMedia*, vol. 21, no. 2, pp. 42–70, 2014.

[85] H. Xu, C. Jiang, X. Liang, and Z. Li, "Spatial-aware graph relation network for large-scale object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 9298–9307.

[86] H. Xu, C. Jiang, X. Liang, L. Lin, and Z. Li, "Reasoning-rcnn: Unifying adaptive global reasoning into large-scale object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 6419–6428.

[87] K. Liang, Y. Guo, H. Chang, and X. Chen, "Visual relationship detection with deep structural ranking," in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[88] Y. Zhan, J. Yu, T. Yu, and D. Tao, "On exploring undetermined relationships for visual relationship detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 5128–5137.

[89] Y. Li, D. Tarlow, M. Brockschmidt, and R. Zemel, "Gated graph sequence neural networks," *arXiv preprint arXiv:1511.05493*, 2015.

[90] Y. Cui, M. Jia, T.-Y. Lin, Y. Song, and S. Belongie, "Class-balanced loss based on effective number of samples," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 9268–9277.

[91] A. Bosselut, H. Rashkin, M. Sap, C. Malaviya, A. Celikyilmaz, and Y. Choi, "Comet: Commonsense transformers for automatic knowledge graph construction," *arXiv preprint arXiv:1906.05317*, 2019.

[92] F. Ilievski, P. Szekely, J. Cheng, F. Zhang, and E. Qasemi, "Consolidating commonsense knowledge," *arXiv preprint arXiv:2006.06114*, 2020.

[93] A. Zareian, Z. Wang, H. You, and S.-F. Chang, "Learning visual commonsense for robust scene graph generation," in *Computer Vision – ECCV 2020*, Springer International Publishing, 2020, pp. 642–657, ISBN: 978-3-030-58592-1.

[94] K. Kumar Singh, S. Divvala, A. Farhadi, and Y. Jae Lee, "Dock: Detecting objects by transferring common-sense knowledge," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 492–508.

[95] C. Jiang, H. Xu, X. Liang, and L. Lin, "Hybrid knowledge routed modules for large-scale object detection," in *Advances in Neural Information Processing Systems*, 2018, pp. 1559–1570.

[96] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, 2017, pp. 5998–6008.

[97] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[98] M. Qi, Y. Wang, J. Qin, and A. Li, "Ke-gan: Knowledge embedded generative adversarial networks for semi-supervised scene parsing," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 5237–5246.

[99] M. Narasimhan and A. G. Schwing, "Straight to the facts: Learning knowledge base retrieval for factual visual question answering," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 451–468.

[100] Z. Su, C. Zhu, Y. Dong, D. Cai, Y. Chen, and J. Li, "Learning visual knowledge memory networks for visual question answering," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7736–7745.

[101] R. Speer, J. Chin, and C. Havasi, "Conceptnet 5.5: An open multilingual graph of general knowledge," in *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

[102] O. Groth, F. B. Fuchs, I. Posner, and A. Vedaldi, "Shapestacks: Learning vision-based physical intuition for generalised object stacking," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 702–717.

[103] C.-Y. Chuang, J. Li, A. Torralba, and S. Fidler, "Learning to act properly: Predicting and explaining affordances from images," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 975–983.

[104] T. Wang, J. Huang, H. Zhang, and Q. Sun, "Visual commonsense r-cnn," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 10 760–10 770.

[105] X. Wang, R. Girshick, A. Gupta, and K. He, "Non-local neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 7794–7803.

[106] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks," *arXiv preprint arXiv:1710.10903*, 2017.

[107] A. Malinin and M. Gales, "Predictive uncertainty estimation via prior networks," in *Advances in Neural Information Processing Systems*, 2018, pp. 7047–7058.

[108] L. Romaszko, C. K. Williams, P. Moreno, and P. Kohli, "Vision-as-inverse-graphics: Obtaining a rich 3d explanation of a scene from a single image," in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 2017, pp. 851–859.

[109] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, "Extracting and composing robust features with denoising autoencoders," in *Proceedings of the 25th international conference on Machine learning*, 2008, pp. 1096–1103.

[110] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[111] G. Alain and Y. Bengio, "What regularized auto-encoders learn from the data-generating distribution," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 3563–3593, 2014.

[112] M. Li, A. Zareian, Q. Zeng, S. Whitehead, D. Lu, H. Ji, and S.-F. Chang, "Cross-media structured common space for multimedia event extraction," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Association for Computational Linguistics, 2020, pp. 2557–2568.

[113] H. Wu, J. Mao, Y. Zhang, Y. Jiang, L. Li, W. Sun, and W.-Y. Ma, "Univse: Robust visual semantic embeddings via structured semantic representations," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[114] F. Faghri, D. J. Fleet, J. R. Kiros, and S. Fidler, "Vse++: Improving visual-semantic embeddings with hard negatives," *Proceedings of the British Machine Vision Conference (BMVC)*, 2018.

[115] A. Karpathy and L. Fei-Fei, "Deep visual-semantic alignments for generating image descriptions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3128–3137.

[116] H. Ji and R. Grishman, "Refining event extraction through cross-document inference," in *Proceedings of ACL-08: HLT*, 2008, pp. 254–262.

[117] S. Liao and R. Grishman, "Acquiring topic features to improve event extraction: In pre-selected and balanced collections," in *Proc. RANLP2011*, 2011.

[118] R. Huang and E. Riloff, "Bootstrapped training of event extraction classifiers," in *Proc. EACL2012*, 2012.

[119] Q. Li, H. Ji, and L. Huang, "Joint event extraction via structured prediction with global features," in *Proc. ACL2013*, 2013.

[120] Y. Chen, L. Xu, K. Liu, D. Zeng, and J. Zhao, "Event extraction via dynamic multi-pooling convolutional neural networks," in *Proc. ACL-IJCNLP2015*, 2015.

[121] T. H. Nguyen, K. Cho, and R. Grishman, "Joint event extraction via recurrent neural networks," in *Proc. NAACL-HLT2016*, 2016.

[122] Y. Hong, W. Zhou, j. zhang jingli, G. Zhou, and Q. Zhu, "Self-regulation: Employing a generative adversarial network to improve event detection," in *Proc. ACL2018*, Melbourne, Australia, 2018.

[123] X. Liu, Z. Luo, and H. Huang, "Jointly multiple events extraction via attention-based graph information aggregation," in *Proc. EMNLP2018*, Brussels, Belgium, 2018.

[124] Y. Chen, H. Yang, K. Liu, J. Zhao, and Y. Jia, "Collective event detection via a hierarchical and bias tagging networks with gated multi-level attention mechanisms," in *Proc. EMNLP2018*, Brussels, Belgium, 2018.

[125] T. Zhang, H. Ji, and A. Sil, "Joint entity and event extraction with generative adversarial imitation learning," *Data Intelligence Vol 1 (2): 99-120*, 2019.

[126] X. Liu, Z. Luo, and H. Huang, "Jointly multiple events extraction via attention-based graph information aggregation," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2018, pp. 1247–1256.

[127] R. Wang, D. Zhou, and Y. He, "Open event extraction from online text using a generative adversarial network," *arXiv preprint arXiv:1908.09246*, 2019.

[128] S. Yang, D. Feng, L. Qiao, Z. Kan, and D. Li, "Exploring pre-trained language models for event extraction and generation," in *Proceedings of the 57th Conference of the Association for Computational Linguistics*, 2019, pp. 5284–5294.

[129] T. Zhang, S. Whitehead, H. Zhang, H. Li, J. Ellis, L. Huang, W. Liu, H. Ji, and S.-F. Chang, "Improving event extraction via multimodal integration," in *Proceedings of the 25th ACM international conference on Multimedia*, ACM, 2017, pp. 270–278.

[130] C. Gu, C. Sun, D. A. Ross, C. Vondrick, C. Pantofaru, Y. Li, S. Vijayanarasimhan, G. Toderici, S. Ricco, R. Sukthankar, *et al.*, "Ava: A video dataset of spatio-temporally localized atomic visual actions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 6047–6056.

[131] F. Caba Heilbron, V. Escorcia, B. Ghanem, and J. Carlos Niebles, "Activitynet: A large-scale video benchmark for human activity understanding," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 961–970.

[132] G. Ye, Y. Li, H. Xu, D. Liu, and S.-F. Chang, "Eventnet: A large scale structured concept library for complex event detection in video," in *Proceedings of the 23rd ACM international conference on Multimedia*, ACM, 2015, pp. 471–480.

[133] D. Li, Z. Qiu, Q. Dai, T. Yao, and T. Mei, "Recurrent tubelet proposal and recognition networks for action detection," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 303–318.

[134] K. Duarte, Y. Rawat, and M. Shah, "Videocapsulenet: A simplified network for action detection," in *Advances in Neural Information Processing Systems*, 2018, pp. 7610–7619.

[135] G. A. Sigurdsson, G. Varol, X. Wang, A. Farhadi, I. Laptev, and A. Gupta, "Hollywood in homes: Crowdsourcing data collection for activity understanding," in *European Conference on Computer Vision*, Springer, 2016, pp. 510–526.

[136] C.-Y. Wu, C. Feichtenhofer, H. Fan, K. He, P. Krahenbuhl, and R. Girshick, "Long-term feature banks for detailed video understanding," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 284–293.

[137] C. Silberer and M. Pinkal, "Grounding semantic roles in images," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2018, pp. 2616–2626.

[138] R. Kiros, R. Salakhutdinov, and R. S. Zemel, "Unifying visual-semantic embeddings with multimodal neural language models," *arXiv preprint arXiv:1411.2539*, 2014.

[139] Y. Liu, Y. Guo, E. M. Bakker, and M. S. Lew, "Learning a recurrent residual fusion network for multimodal matching," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 4107–4116.

[140] L. Wang, Y. Li, and S. Lazebnik, "Learning deep structure-preserving image-text embeddings," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 5005–5013.

[141] A. Karpathy, A. Joulin, and L. F. Fei-Fei, "Deep fragment embeddings for bidirectional image sentence mapping," *Advances in neural information processing systems*, vol. 27, pp. 1889–1897, 2014.

[142] F. Xiao, L. Sigal, and Y. Jae Lee, "Weakly-supervised visual grounding of phrases with linguistic structures," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 5945–5954.

[143] S. Datta, K. Sikka, A. Roy, K. Ahuja, D. Parikh, and A. Divakaran, "Align2ground: Weakly supervised phrase grounding guided by image-caption alignment," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 2601–2610.

[144] K. Chen, J. Gao, and R. Nevatia, "Knowledge aided consistency for weakly supervised phrase grounding," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4042–4050.

[145] H. Akbari, S. Karaman, S. Bhargava, B. Chen, C. Vondrick, and S.-F. Chang, "Multi-level multimodal common semantic space for image-phrase grounding," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 12 476–12 486.

[146] W. Su, X. Zhu, Y. Cao, B. Li, L. Lu, F. Wei, and J. Dai, "Vl-bert: Pre-training of generic visual-linguistic representations," *arXiv preprint arXiv:1908.08530*, 2019.

[147] L. H. Li, M. Yatskar, D. Yin, C.-J. Hsieh, and K.-W. Chang, "Visualbert: A simple and performant baseline for vision and language," *arXiv preprint arXiv:1908.03557*, 2019.

[148] Y.-C. Chen, L. Li, L. Yu, A. E. Kholy, F. Ahmed, Z. Gan, Y. Cheng, and J. Liu, "Uniter: Learning universal image-text representations," *arXiv preprint arXiv:1909.11740*, 2019.

[149] C. J. Fillmore, C. R. Johnson, and M. R. Petruck, "Background to framenet," *International journal of lexicography*, vol. 16, no. 3, pp. 235–250, 2003.

[150] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[151] L. Banarescu, C. Bonial, S. Cai, M. Georgescu, K. Griffitt, U. Hermjakob, K. Knight, P. Koehn, M. Palmer, and N. Schneider, "Abstract meaning representation for sembanking," in *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*, 2013, pp. 178–186.

[152] C. Wang, N. Xue, and S. Pradhan, "A transition-based algorithm for amr parsing," in *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Denver, Colorado: Association for Computational Linguistics, 2015, pp. 366–375.

[153] ——, "Boosting transition-based amr parsing with refined actions and auxiliary analyzers," in *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, Beijing, China: Association for Computational Linguistics, 2015, pp. 857–862.

[154] C. Wang, S. Pradhan, X. Pan, H. Ji, and N. Xue, "Camr at semeval-2016 task 8: An extended transition-based amr parser," in *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, San Diego, California: Association for Computational Linguistics, 2016, pp. 1173–1178.

[155] C. D. Manning, M. Surdeanu, J. Bauer, J. Finkel, S. J. Bethard, and D. McClosky, "The Stanford CoreNLP natural language processing toolkit," in *Association for Computational Linguistics (ACL) System Demonstrations*, 2014, pp. 55–60.

[156] A. Graves, A.-r. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *2013 IEEE international conference on acoustics, speech and signal processing*, IEEE, 2013, pp. 6645–6649.

[157] C. Walker, S. Strassel, J. Medero, and K. Maeda, "Ace 2005 multilingual training corpus," *Linguistic Data Consortium, Philadelphia*, vol. 57, 2006.

[158] H. Shi, J. Mao, T. Xiao, Y. Jiang, and J. Sun, "Learning visually-grounded semantics from contrastive adversarial samples," *arXiv preprint arXiv:1806.10348*, 2018.

[159] A. Zareian, K. D. Rosa, D. H. Hu, and S.-F. Chang, "Open-vocabulary object detection using captions," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021.

[160] A. Kuznetsova, H. Rom, N. Alldrin, J. Uijlings, I. Krasin, J. Pont-Tuset, S. Kamali, S. Popov, M. Malloci, A. Kolesnikov, *et al.*, "The open images dataset v4," *International Journal of Computer Vision*, pp. 1–26, 2020.

[161] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *European conference on computer vision*, Springer, 2014, pp. 740–755.

[162] A. Frome, G. S. Corrado, J. Shlens, S. Bengio, J. Dean, M. Ranzato, and T. Mikolov, "Devise: A deep visual-semantic embedding model," in *Advances in neural information processing systems*, 2013, pp. 2121–2129.

[163] S. Rahman, S. Khan, and N. Barnes, "Improved visual-semantic alignment for zero-shot object detection," *34th AAAI Conference on Artificial Intelligence*, 2020.

[164] P. Zhu, H. Wang, and V. Saligrama, "Zero shot detection," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 30, no. 4, pp. 998–1010, 2019.

[165] ——, "Don't even look once: Synthesizing features for zero-shot detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 11 693–11 702.

[166] Z. Li, L. Yao, X. Zhang, X. Wang, S. Kanhere, and H. Zhang, "Zero-shot object detection with textual descriptions," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, 2019, pp. 8690–8697.

[167] S. Rahman, S. Khan, and N. Barnes, "Transductive learning for zero-shot object detection," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 6082–6091.

[168] D. Gupta, A. Anantharaman, N. Mamgain, V. N. Balasubramanian, C. Jawahar, *et al.*, "A multi-space approach to zero-shot object detection," in *The IEEE Winter Conference on Applications of Computer Vision*, 2020, pp. 1209–1217.

[169] F. Wan, C. Liu, W. Ke, X. Ji, J. Jiao, and Q. Ye, "C-mil: Continuation multiple instance learning for weakly supervised object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 2199–2208.

[170] R. G. Cinbis, J. Verbeek, and C. Schmid, "Weakly supervised object localization with multi-fold multiple instance learning," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 1, pp. 189–203, 2016.

[171] K. Ye, M. Zhang, A. Kovashka, W. Li, D. Qin, and J. Berent, "Cap2det: Learning to amplify weak caption supervision for object detection," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 9686–9695.

[172] E. Amrani, R. Ben-Ari, T. Hakim, and A. Bronstein, "Learning to detect and retrieve objects from unlabeled videos," in *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, IEEE, 2019, pp. 3713–3717.

[173] C. Sun, C. Gan, and R. Nevatia, "Automatic concept discovery from parallel text and visual corpora," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 2596–2604.

[174] K. Ye, M. Zhang, W. Li, D. Qin, A. Kovashka, and J. Berent, "Learning to discover and localize visual objects with open vocabulary," *arXiv preprint arXiv:1811.10080*, 2018.

[175] J. Gao, J. Wang, S. Dai, L.-J. Li, and R. Nevatia, "Note-rcnn: Noise tolerant ensemble rcnn for semi-supervised object detection," in *Proceedings of the IEEE international conference on computer vision*, 2019, pp. 9508–9517.

[176] V. Ramanathan, R. Wang, and D. Mahajan, "Dlwl: Improving detection for lowshot classes with weakly labelled data," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 9342–9352.

[177] Y.-X. Wang and M. Hebert, "Model recommendation: Generating object detectors from few samples," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1619–1628.

[178] J. Hoffman, S. Guadarrama, E. S. Tzeng, R. Hu, J. Donahue, R. Girshick, T. Darrell, and K. Saenko, "Lsda: Large scale detection through adaptation," in *Advances in Neural Information Processing Systems*, 2014, pp. 3536–3544.

[179] Y. Tang, J. Wang, B. Gao, E. Dellandréa, R. Gaizauskas, and L. Chen, "Large scale semi-supervised object detection using visual and semantic knowledge transfer," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2119–2128.

[180] J. Uijlings, S. Popov, and V. Ferrari, "Revisiting knowledge transfer for training object class detectors," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 1101–1110.

[181] K. Desai and J. Johnson, "Virtex: Learning visual representations from textual annotations," *arXiv preprint arXiv:2006.06666*, 2020.

[182] L. H. Li, H. You, Z. Wang, A. Zareian, S.-F. Chang, and K.-W. Chang, "Weakly-supervised visualbert: Pre-training without parallel images and captions," *arXiv preprint arXiv:2010.12831*, 2020.

[183] Y.-C. Chen, L. Li, L. Yu, A. E. Kholy, F. Ahmed, Z. Gan, Y. Cheng, and J. Liu, "Uniter: Universal image-text representation learning," in *ECCV*, 2020.

[184] W. Su, X. Zhu, Y. Cao, B. Li, L. Lu, F. Wei, and J. Dai, "Vl-bert: Pre-training of generic visual-linguistic representations," in *International Conference on Learning Representations*, 2020.

[185] Z. Huang, Z. Zeng, B. Liu, D. Fu, and J. Fu, "Pixel-bert: Aligning image pixels with text by deep multi-modal transformers," *arXiv preprint arXiv:2004.00849*, 2020.

[186] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*, Ieee, 2009, pp. 248–255.

[187] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[188] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2961–2969.

[189] X. Chen, H. Fang, T.-Y. Lin, R. Vedantam, S. Gupta, P. Dollár, and C. L. Zitnick, "Microsoft coco captions: Data collection and evaluation server," *arXiv preprint arXiv:1504.00325*, 2015.

[190] P. Sharma, N. Ding, S. Goodman, and R. Soricut, "Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2018, pp. 2556–2565.

[191] F. Massa and R. Girshick, *maskrcnn-benchmark: Fast, modular reference implementation of Instance Segmentation and Object Detection algorithms in PyTorch*, `https://github.com/facebookresearch/maskrcnn-benchmark`, Accessed: [Insert date here], 2018.

[192] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, J. Davison, S. Shleifer, P. von Platen, C. Ma, Y. Jernite, J. Plu, C. Xu, T. L. Scao, S. Gugger, M. Drame, Q. Lhoest, and A. M. Rush, "Huggingface's transformers: State-of-the-art natural language processing," *ArXiv*, vol. abs/1910.03771, 2019.

[193] C. L. Zitnick and P. Dollár, "Edge boxes: Locating object proposals from edges," in *European conference on computer vision*, Springer, 2014, pp. 391–405.

[194] L. van der Maaten and G. Hinton, "Visualizing data using t-sne," *Journal of Machine Learning Research*, vol. 9, no. 86, pp. 2579–2605, 2008.

[195] X. Shang, T. Ren, J. Guo, H. Zhang, and T.-S. Chua, "Video visual relation detection," in *Proceedings of the 25th ACM international conference on Multimedia*, ACM, 2017, pp. 1300–1308.

[196] Y.-H. H. Tsai, S. Divvala, L.-P. Morency, R. Salakhutdinov, and A. Farhadi, "Video relationship reasoning using gated spatio-temporal energy graph," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 10 424–10 433.

[197] J. Ji, R. Krishna, L. Fei-Fei, and J. C. Niebles, "Action genome: Actions as compositions of spatio-temporal scene graphs," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 10 236–10 247.

[198] A. Miech, J.-B. Alayrac, L. Smaira, I. Laptev, J. Sivic, and A. Zisserman, "End-to-end learning of visual representations from uncurated instructional videos," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 9879–9889.

[199] C. Gan, C. Sun, and R. Nevatia, "Deck: Discovering event composition knowledge from web images for zero-shot event detection and recounting in videos," in *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

[200] A. Kukleva, M. Tapaswi, and I. Laptev, "Learning interactions and relationships between movie characters," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 9849–9858.

[201] X. Yang, K. Tang, H. Zhang, and J. Cai, "Auto-encoding scene graphs for image captioning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 10 685–10 694.

[202] D. A. Hudson and C. D. Manning, "Learning by abstraction: The neural state machine," *arXiv preprint arXiv:1907.03950*, 2019.

[203] H.-J. Ehni, "Dual use and the ethical responsibility of scientists," *Archivum immunologiae et therapiae experimentalis*, vol. 56, no. 3, p. 147, 2008.

[204] D. Hovy and S. L. Spruit, "The social impact of natural language processing," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, Berlin, Germany: Association for Computational Linguistics, Aug. 2016, pp. 591–598.

[205] M. Brundage, S. Avin, J. Clark, H. Toner, P. Eckersley, B. Garfinkel, A. Dafoe, P. Scharre, T. Zeitzoff, B. Filar, H. Anderson, H. Roff, G. C. Allen, J. Steinhardt, C. Flynn, S. O. hEigeartaigh, S. Beard, H. Belfield, S. Farquhar, C. Lyle, R. Crootof, O. Evans, M. Page, J. Bryson, R. Yampolskiy, and D. Amodei, "The malicious use of artifcial intelligence: Forecasting, prevention, and mitigation," *arXiv:1802.07228*, 2018.

[206] N. Akhtar and A. Mian, "Threat of adversarial attacks on deep learning in computer vision: A survey," *IEEE Access*, vol. 6, pp. 14 410–14 430, 2018.

[207] J. Buolamwini and T. Gebru, "Gender shades: Intersectional accuracy disparities in commercial gender classification," in *Conference on fairness, accountability and transparency*, 2018, pp. 77–91.

[208] R. Rudinger, J. Naradowsky, B. Leonard, and B. Van Durme, "Gender bias in coreference resolution," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, 2018, pp. 8–14.